

This Internet-Draft, draft-bajko-pripaddrassign-03.txt, has expired, and has been deleted from the Internet-Drafts directory. An Internet-Draft expires 185 days from the date that it is posted unless it is replaced by an updated version, or the Secretariat has been notified that the document is under official review by the IESG or has been passed to the RFC Editor for review and/or publication as an RFC. This Internet-Draft was not published as an RFC.

Internet-Drafts are not archival documents, and copies of Internet-Drafts that have been deleted from the directory are not available. The Secretariat does not have any information regarding the future plans of the authors or working group, if applicable, with respect to this deleted Internet-Draft. For more information, or to request a copy of the document, please contact the authors directly.

Draft Authors:

Gabor Bajko<Gabor.Bajko@nokia.com>

Teemu Savolainen<teemu.savolainen@nokia.com>

Mohammed Boucadair<mohamed.boucadair@orange-ftgroup.com>

Pierre Levis<pierre.levis@orange-ftgroup.com>

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 18, 2013

S. Bhandari
S. Gundavelli
Cisco Systems
J. Korhonen
Renesas Mobile
M. Grayson
Cisco Systems
April 16, 2013

Access-Network-Identifier Option in DHCP
draft-bhandari-dhc-access-network-identifier-04

Abstract

This document specifies the format and mechanism that is to be used for encoding access network identifiers in DHCPv4 and DHCPv6 messages by defining new access network identifier options and sub-options.

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 October 18, 2013.

Copyright Notice

Copyright (c) 2013 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. Motivation	3
3. Terminology	4
4. DHCPv4 Access-Network-Identifier Option	5
4.1. DHCPv4 Access-Network-Identifier Sub-options	5
5. DHCPv6 Access-Network-Identifier options	6
6. DHCPv4 and DHCPv6 Access-Network-Identifier Options	6
6.1. Access-Network-Type option	6
6.2. Network-Identifier options	8
6.3. Operator identifier options	11
7. Client Behavior	12
8. Relay Agent Behavior	13
9. Server Behavior	13
10. IANA Considerations	13
11. Security Considerations	14
12. Acknowledgements	14
13. Change log	14
14. Normative References	15
Authors' Addresses	16

1. Introduction

Access network identification of a network device has a range of application. For e.g. The local mobility anchor in a Proxy Mobile IPv6 domain is able to provide access network and access operator specific handling or policing of the mobile node traffic using information about the access network to which the mobile node is attached.

This document specifies Dynamic Host Configuration Protocol v4 (DHCPv4) [RFC2131] and Dynamic Host Configuration Protocol v6 (DHCPv6) [RFC3315] options for access network identification that is added by Client or Relay agent in the DHCPv4 or DHCPv6 messages towards the Server.

Dynamic Host Configuration Protocol (DHCP) client or DHCP relay agent aware of the access network and access operator add this information in the DHCP messages. This information can be used to provide differentiated services and policing of traffic based on the access network to which a client is attached. Examples of how this information can be used in mobile networks can be found in [RFC6757]

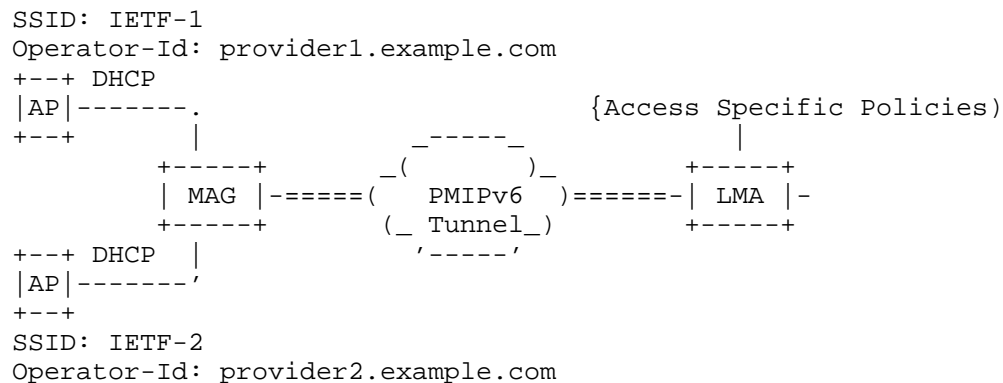
2. Motivation

Proxy mobile IPv6 [RFC5213] can be used for supporting network-based mobility management in various type of network deployments. The network architectures, such as Service provider Wi-Fi access aggregation or, WLAN integrated mobile packet core are examples where Proxy Mobile IPv6 is a component of the overall architecture. Some of these architectures require the ability of the local mobility anchor (LMA) [RFC5213] to provide differentiated services and policing of traffic to the mobile nodes based on the access network to which they are attached. Policy systems in mobility architectures such as PCC [TS23203] and ANDSF [TS23402] in 3GPP system allow configuration of policy rules with conditions based on the access network information. For example, the service treatment for the mobile node's traffic may be different when they are attached to a access network owned by the home operator than when owned by a roaming partner. The service treatment can also be different based on the configured Service Set Identifiers (SSID) in case of IEEE 802.11 based access networks. Other examples of services include the operator's ability to apply tariff based on the location.

The PMIPv6 extension as specified in [RFC6757] defines PMIPv6 options to carry access network identifiers in PMIPv6 signaling from Mobile Access Gateway (MAG) to LMA. MAG can learn this information from DHCP options as inserted by DHCP client or Relay agent before MAG.

If MAG relays DHCP messages to LMA as specified in [RFC5844] this information can be inserted by MAG towards LMA in the forwarded DHCP messages.

Figure 1 illustrates an example Proxy Mobile IPv6 deployment where Access Points (AP) inserts access network identifiers in DHCP messages. The mobile access gateway learns this information over DHCP and delivers the information elements related to the access network to the local mobility anchor over Proxy Mobile IPv6 signaling messages. In this example, the additional information could comprise the SSID of the used IEEE 802.11 network and the identities of the operators running the IEEE 802.11 access network infrastructure.



Access Networks attached to MAG

3. Terminology

All the DHCP related terms used in this document to be interpreted as defined in the Dynamic Host Configuration Protocol v4 (DHCPv4) [RFC2131] and Dynamic Host Configuration Protocol v6 (DHCPv6) [RFC3315] specifications. DHCP refers to both DHCPv4 and DHCPv6 messages and entities throughout this document.

All the mobility related terms used in this document are to be interpreted as defined in the Proxy Mobile IPv6 specifications [RFC5213] and [RFC5844]. Additionally, this document uses the following abbreviations:

Service Set Identifier Service Set Identifier (SSID) identifies the name of the IEEE 802.11 network. SSID differentiates from one network to the other.

Vendor ID The Vendor ID is the SMI Network Management Private Enterprise Code of the IANA-maintained Private Enterprise Numbers registry [SMI].

4. DHCPv4 Access-Network-Identifier Option

Access network identifier option carries information to identify the access network to which the client is attached to. This information includes access technology type, network identifier and access network operator identifiers.

The format of the DHCPv4 Access-Network-Identifier option is shown below.

Code	Len	ANI Sub-options				
code	len	s1	s2	s2	...	sn

code: 8-bit code carrying Access Network Identifier sub-options,
 If added by relay agent: Relay Agent Information Option (82)
 If added by client: OPTION_ACCESS_NETWORK_ID (TBD1)

len: 8 bit indicating total length of the included suboptions.

ANI Sub-options: The ANI Sub-options consists of a sequence of SubOpt/Length/Value tuples for each sub-option, encoded in the following manner:

SubOpt	Len	Sub-option Value				
code	N	s1	s2	s3	s4	sN

ANI Sub-options are defined in following sections.

4.1. DHCPv4 Access-Network-Identifier Sub-options

Access network identifier information will be defined in multiple sub-options. The initial assignment of DHCP access network identifier Sub-options is as follows:

Sub-option Code	Sub-Option Description
TBD7	Access-Network-Type Sub-option
TBD8	Network-Name Sub-option
TBD9	AP-Name Sub-option
TBD10	Operator-Identifier Sub-option
TBD11	Operator-Realm Sub-option

5. DHCPv6 Access-Network-Identifier options

The Access Network Identifier option defined here will be added by DHCPv6 client in upstream DHCPv6 messages or by the Relay in Relay-forward messages.

Option Code	Description
TBD2	OPTION_ANI_ATT
TBD3	OPTION_ANI_NETWORK_NAME
TBD4	OPTION_ANI_AP_NAME
TBD5	OPTION_ANI_OPERATOR_ID
TBD6	OPTION_ANI_OPERATOR_REALM

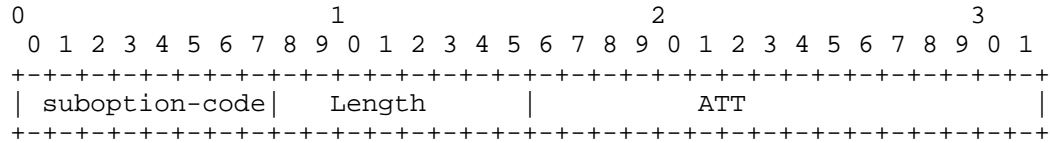
6. DHCPv4 and DHCPv6 Access-Network-Identifier Options

This section defines DHCPv4 suboption and DHCPv6 options for access network identification.

6.1. Access-Network-Type option

This option is used for exchanging the type of the access technology the client is attached to the network. There can only be a single instance of this specific option in any DHCPv6 message or single instance of this specific sub-option in DHCPv4 OPTION_ACCESS_NETWORK_ID or Relay Agent information option. Its format is as follows:

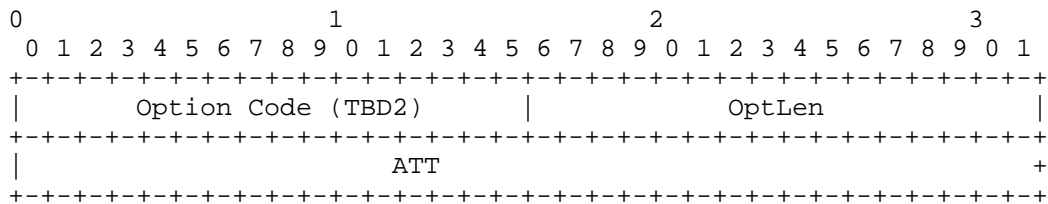
DHCPv4:



suboption-code: 8-bit code, it should be set to value of (TBD7),
indicating that its a Access-Network-Type sub-option

Length: 8-bit unsigned integer indicating the length of this suboption
in octets, excluding the suboption-code and length fields.
This field MUST be set to 2.

DHCPv6:



option-code: 16-bit code OPTION_ANI_ATT (TBD2)
option-length: 16-bit unsigned integer indicating length
in octets of this option

Common format applicable to DHCPv4 and DHCPv6:
Access Technology Type (ATT)

An 16-bit field that specifies the access technology through
which the client is connected to the access link.

The values is as populated from the IANA name space
Access Technology Type Option type values as requested in [RFC5213]

0: Reserved	("Reserved")
1: Virtual	("Logical Network Interface")
2: PPP	("Point-to-Point Protocol")
3: IEEE 802.3	("Ethernet")
4: IEEE 802.11a/b/g	("Wireless LAN")
5: IEEE 802.16e	("WIMAX")

6.2. Network-Identifier options

These options can be used for carrying the name of the access network (e.g., a SSID in case of IEEE 802.11 Access Network, or PLMN Identifier [TS23003] in case of 3GPP access) and Access Point name, to which the client is attached. There can only be a single instance of each of these options in any DHCPv6 message or single instance of each of these sub-options in DHCPv4 `OPTION_ACCESS_NETWORK_ID` or Relay Agent information option. The format of these options is defined below.

DHCPv4:

```

      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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|suboption code |   Length   |                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Network Name (e.g., SSID or PLMNID) ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

suboption code: 8-bit code, it should be set to value of (TBD8), indicating that its a Network-Name sub-option

Length: 8-bit indicating Total length of this sub option,
excluding the suboption code and length fields.
The value can be in the range of 2 to 32 octets.

DHCPv6:

```

      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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Option Code (TBD3)          |          OptLen          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Network Name (e.g., SSID or PLMNID) ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

option-code: 16-bit code `OPTION_ANI_NETWORK_NAME` (TBD3)
option-length: 16-bit unsigned integer indicating length
in octets of this option. The value can be in the
range of 2 to 32 octets.

Common format applicable to DHCPv4 and DHCPv6:

Network Name: The name of the access network to which the mobile node is attached. The type of the Network Name is dependent on the access technology to which the mobile node is attached. If it is 802.11 access, the Network Name MUST be the SSID of the network. If the access network is 3GPP access, the Network Name is the PLMN Identifier of the network. If the access network is 3GPP2 access, the Network Name is the Access Network Identifier [ANI].

When encoding the PLMN Identifier, both the Mobile Network Code (MNC) [TS23003] and Mobile Country Code (MCC) [TS23003] MUST be 3 digits. If the MNC in use only has 2 digits, then it MUST be preceded with a '0'. Encoding MUST be UTF-8.

DHCPv4:

```

      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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|suboption code |   Length   |                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Access-Point Name             ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

suboption code: 8-bit code, it should be set to value of (TBD9),
indicating that its a Network-AP-Name sub-option

Length: 8-bit indicating Total length of this sub option,
excluding the suboption code and length fields.
The value can be in the range of 2 to 32 octets.

DHCPv6:

```

      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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Option Code (TBD3)          |          OptLen          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Access-Point Name             ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

option-code: 16-bit code OPTION_ANI_AP_NAME (TBD4)
option-length: 16-bit unsigned integer indicating length
in octets of this option. The value can be in the
range of 2 to 32 octets.

Common format applicable to DHCPv4 and DHCPv6:

Access-Point Name: The name of the access point (physical device name) to which the mobile node is attached. This is the identifier that uniquely identifies the access point. While Network Name (e.g., SSID) identifies the operator's access network, Access-Point Name identifies a specific network device in the network to which the mobile node is attached. In some deployments, the Access-Point Name can be set to the Media Access Control (MAC) address of the device or some unique identifier that can be used by the policy systems in the operator network to unambiguously identify the device. The string is carried in UTF-8 representation.

6.3. Operator identifier options

The Operator identifier options can be used for carrying the operator identifier of the access network to which the client is attached. There can only be a single instance of each of these options in any DHCPv6 message or single instance of each of these sub-options in DHCPv4 `OPTION_ACCESS_NETWORK_ID` or Relay Agent information option. The format of these options is defined below.

DHCPv4:

```

      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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| suboptioncode |      Length      |                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+
~      Operator Enterprise ID      |
+-----+-----+-----+-----+-----+-----+

```

suboption code: 8 bit code, It should be set to value of (TBD10), indicating that it is Operator-Identifier sub-option

Length: Total length of this sub option, excluding the suboption code and length fields.

DHCPv6:

```

      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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Option Code (TBD4)      |      OptLen      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Operator Enterprise ID      |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

option-code: 16-bit code `OPTION_ANI_OPERATOR_ID` (TBD5)

option-length: 16-bit unsigned integer indicating length in octets of this option.

Common format applicable to DHCPv4 and DHCPv6:

Operator Enterprise ID: Vendor ID as a four octet
Private Enterprise Number [SMI].

DHCPv4:

```

      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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| suboptioncode |      Length      |                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+
~                                     Operator Realm                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

suboption code: 8 bit code, It should be set to value of (TBD11), indicating that it is Operator-Realm sub-option

Length: Total length of this sub option, excluding the suboption code and length fields.

DHCPv6:

```

      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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Option Code (TBD4)      |      OptLen      |
+-----+-----+-----+-----+-----+-----+-----+-----+
~                                     Operator Realm                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

option-code: 16-bit code OPTION_ANI_OPERATOR_REALM (TBD6)
option-length: 16-bit unsigned integer indicating length
in octets of this option.

Common format applicable to DHCPv4 and DHCPv6:

Operator Realm: Realm of the operator. Realm names are required to be unique, and are piggybacked on the administration of the DNS namespace. Realms are encoded using a domain name encoding defined in [RFC1035]. Up to 253 octets of the operator realm.

7. Client Behavior

All hosts or clients MAY include access network identifier options in all the upstream DHCP messages to inform the receiver about the access network it is attached to.

8. Relay Agent Behavior

DHCP Relay Agents MAY include these options before forwarding the DHCP message to provide information about the access network over which DHCP messages from the client is received.

9. Server Behavior

If DHCP Server is unable to understand this option it MUST be ignored. There is no requirement that a server return this option and its data in a downstream DHCP message. If DHCP Server is able to process these options it MAY use it for address pool selection policy decisions if configured. It MAY store this information along with the lease for logging and audit purpose.

10. IANA Considerations

This document defines DHCPv4 Access Network Identifier option which requires assignment of DHCPv4 option code TBD1 assigned from "Bootp and DHCP options" registry (<http://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xml>), as specified in [RFC2939].

IANA is requested to assign Sub-option codes for the following DHCPv4 Sub-options from the "DHCP Relay Agent Sub-Option Codes"

Sub-option Code	Sub-Option Description
-----	-----
TBD7	Access-Network-Type Sub-option
TBD8	Network-Name Sub-option
TBD9	AP-Name Sub-option
TBD10	Operator-Identifier Sub-option
TBD11	Operator-Realm Sub-option

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

Option Code	Description
TBD2	OPTION_ANI_ATT
TBD3	OPTION_ANI_NETWORK_NAME
TBD4	OPTION_ANI_AP_NAME
TBD5	OPTION_ANI_OPERATOR_ID
TBD6	OPTION_ANI_OPERATOR_REALM

11. Security Considerations

Since there is no privacy protection for DHCP messages, an eavesdropper who can monitor the link between the DHCP server, relay agent and client can discover access network information.

To minimize the unintended exposure of this information, this option SHOULD be included by DHCP entities only when it is configured. Where critical decisions might be based on the value of this option, DHCP authentication as defined in "Authentication for DHCP Messages" [RFC3118] and "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)" [RFC3315] SHOULD be used to protect the integrity of the DHCP options. Link-layer confidentiality and integrity protection may also be employed to reduce the risk of disclosure and tampering.

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

12. Acknowledgements

The authors would like to thank Kim Kinnear, Ted Lemon, Gaurav Halwasia, Bernie Volz for their valuable inputs.

13. Change log

Changes from 00 - 01

- o Modified v4 top level option to be either option 82 if added by relay or a new top level option if added by client
- o Removed DHCPv6 container option
- o Reorganized the options to converge v4 and v6 option descriptions

Changes from 01-02

- o Modified v4 DHCP option format to align with the 1 byte code, len
- o Corrected typos

Changes from 02-03

- o No change

Changes from 03-04

- o split network name and ap name into separate options, removed E bit allowing different encoding
- o corrected the option code, type alignment to match the boundary
- o split operator id into enterprise id and realm as separate options

14. Normative References

- [ANI] "Interoperability Specification (IOS) for High Rate Packet Data (HRPD) Radio Access Network Interfaces with Session Control in the Access Network, A.S0008-A v3.0", October 2008.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.
- [RFC2939] Droms, R., "Procedures and IANA Guidelines for Definition of New DHCP Options and Message Types", BCP 43, RFC 2939, September 2000.
- [RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC5213] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.

- [RFC5844] Wakikawa, R. and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6", RFC 5844, May 2010.
- [RFC6757] Gundavelli, S., Korhonen, J., Grayson, M., Leung, K., and R. Pazhyannur, "Access Network Identifier (ANI) Option for Proxy Mobile IPv6", RFC 6757, October 2012.
- [SMI] "PRIVATE ENTERPRISE NUMBERS, SMI Network Management Private Enterprise Codes", February 2011.
- [TS23003] "Numbering, addressing and identification", 2011.
- [TS23203] "Policy and Charging Control Architecture", 2012.
- [TS23402] "Architecture enhancements for non-3GPP accesses", 2012.

Authors' Addresses

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

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

Email: sgundave@cisco.com

Jouni Korhonen
Renesas Mobile
Linnoitustie 6
FIN-02600 Espoo,
Finland

Phone:
Email: jouni.nospam@gmail.com

Mark Grayson
Cisco Systems
11 New Square Park
Bedfont Lakes, FELTHAM TW14 8HA
ENGLAND

Email: mgrayson@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 16, 2014

S. Bhandari
G. Halwasia
S. Gundavelli
Cisco Systems
H. Deng
China Mobile
L. Thiebaut
Alcatel-Lucent
J. Korhonen
Renesas Mobile
I. Farrer
Deutsche Telekom AG
July 15, 2013

DHCPv6 class based prefix
draft-bhandari-dhc-class-based-prefix-05

Abstract

This document introduces options to communicate property and associate meta data with prefixes. It extends DHCPv6 prefix delegation and address allocation using the meta data for selection of prefixes and addresses.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 16, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
1.1. Motivation	3
1.1.1. Mobile networks	4
1.1.2. Home networks	4
1.2. Terminology	5
1.3. Requirements Language	5
2. Overview	5
2.1. Prefix Property and Class Options	5
2.2. Consideration for different DHCPv6 entities	6
2.2.1. Requesting Router Behavior for IA_PD allocation	7
2.2.2. Delegating Router Behavior for IA_PD allocation	8
2.2.3. DHCPv6 Client Behavior for IA_NA allocation	9
2.2.4. DHCPv6 Server Behavior for IA_NA allocation	9
2.3. Usage	10
2.3.1. Class based prefix and IA_NA allocation	10
2.3.2. Class based prefix and IA_PD allocation	10
2.3.3. Class based prefix and SLAAC	10
2.3.4. Class based prefix and applications	11
3. Example Application	11
3.1. Mobile gateway example	11
3.1.1. Class based prefix delegation	13
3.1.2. IPv6 address assignment from class based prefix	13
3.2. Homenet Example	14
3.2.1. Class based prefix delegation to the HGW	15
3.2.2. IPv6 Assignment to Homenet hosts using stateful DHCPv6	16
4. Acknowledgements	17
5. Contributors	17
6. IANA Considerations	17
6.1. OPTION_PREFIX_PROPERTY values	17
7. Security Considerations	18
8. Change History (to be removed prior to publication as an RFC)	18
9. References	19
9.1. Normative References	19
9.2. Informative References	20
Authors' Addresses	20

1. Introduction

In IPv6 a network interface can acquire multiple addresses from the same scope. In such a multi-prefix network each of the multiple prefixes can have a specific property and purpose associated with it. Example: In a mobile network a mobile device can be assigned a prefix from its home network and another from the visiting network that it is attached to. Another example is a prefix may provide free Internet access without offering any quality of service guarantees while another prefix may be charged along with providing quality of service guarantees for network service access. A prefix can have well defined properties that is universal and have a meta data associated with it that communicates its local significance. The properties and meta data of prefix will be relevant for prefix delegation, source address selection as elaborated in the subsequent sections.

This document defines `OPTION_PREFIX_PROPERTY` option that communicates property of the prefix that is universally understood. This document defines `OPTION_PREFIX_CLASS` option to communicate meta data of the prefix that communicates the prefix's local significance.

This document discusses usage of `OPTION_PREFIX_CLASS` to request and select prefixes with specific meta data via `IA_PD` and `IA_NA` as defined in [RFC3633] and [RFC3315] respectively. This document defines the behavior of the DHCPv6 server, the DHCPv6 prefix requesting router and the DHCPv6 client to use `OPTION_PREFIX_CLASS` option for requesting and selecting prefixes and addresses.

The network address can be configured via DHCPv6 as defined in [RFC3315] or via Stateless Address Autoconfiguration (SLAAC) as defined in [RFC4862], additional information of a prefix can be provided via DHCPv6 or via IPv6 Router Advertisement (RA). The information provided in the options defined in this document `OPTION_PREFIX_PROPERTY` and `OPTION_PREFIX_CLASS` can be used for source address selection. Communicated property and meta data information about the prefix via IPv6 Router Advertisement (RA) will be dealt with in separate document [I-D.korhonen-6man-prefix-properties].

1.1. Motivation

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

1.1.1.1. Mobile networks

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 meta data 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. Neither is a mobile device able to learn about the property of the prefix assigned to influence source address selection. Example to determine if the prefix is a home address or care of address.

1.1.1.2. Home networks

In a fixed network environment, the homenet CPE may also function as both a DHCPv6 client (requesting the IA_PD(s)) and a DHCPv6 server allocating prefixes from delegated prefix(es) to downstream home network hosts. Some service providers may wish to delegate multiple prefixes to the CPE for use by different services classes and traffic types.

Motivations for this include:

- o Using source prefix to identify the service class / traffic type that is being transported. The source prefix may then reliably be used as a parameter for differentiated services or other purposes. E.g. [I-D.jiang-v6ops-semantic-prefix]

- o Using the specific source prefix as a host identifier for other services. E.g. as an input parameter to a DHCPv4 over IPv6 server [I-D.ietf-dhc-dhcpv4-over-ipv6]

To meet these requirements, when the CPE (functioning as a DHCPv6 server) receives an IA_NA or IA_TA request from a homenet host, a mechanism is required so that the correct prefix for requested service class can be selected for allocation. Likewise for DHCPv6 clients located in the homenet, a mechanism is necessary so that the intended service class for a requested prefix can be signalled to the DHCPv6 server.

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 property and prefix class options in IA_PD and IA_NA. This section defines the behavior of the delegating router, the requesting router and the DHCPv6 client. It discusses these options in the context of a DHCPv6 information request from a DHCPv6 client to a DHCPv6 server.

2.1. Prefix Property and Class Options

The format of the DHCPv6 prefix property and prefix class options are 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 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      OPTION_PREFIX_PROPERTY      |      option-length(2)      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Properties      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

option-code:      OPTION_PREFIX_PROPERTY (TBD1)
option-length:    2
Properties:       16 bits maintained as
                  OPTION_PREFIX_PROPERTY in
                  IANA registered namespace.
                  Each value in the registry represents a property.
                  Multiple properties can be represented by bitwise
                  ORing of the individual property values in this
                  field.

```

Prefix Property Option

The individual property are maintained in OPTION_PREFIX_PROPERTY values enumeration explained in Section Section 6.1.

Along with the OPTION_PREFIX_PROPERTY a meta data associated with the prefix that is of local relevance is communicated using OPTION_PREFIX_CLASS defined 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 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      OPTION_PREFIX_CLASS      |      option-length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Prefix Class      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

option-code:      OPTION_PREFIX_CLASS (TBD2)
option-length:    2
Prefix Class:     16 bit integer with the integer value
                  of local significance.

```

Prefix Class Option

2.2. Consideration for different DHCPv6 entities

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

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

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

2.2.1. Requesting Router Behavior for IA_PD allocation

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. The class of prefix it requests is learnt via configuration or any other out of band mechanism not defined in this document.
- 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.

The requesting router parses and stores `OPTION_PREFIX_PROPERTY` if received with the prefix.

2.2.2. Delegating Router Behavior for `IA_PD` allocation

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 classes. Along with including the `IA_PD` prefix options in the `IA_PD` option, it **MAY** include the `OPTION_PREFIX_CLASS` option in the `OPTION_IAPREFIX` option area of the corresponding `IA_PD` prefix option with the class information of the prefix.

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.

In addition if the delegating router has additional property associated with the prefix it will be provided in `OPTION_PREFIX_PROPERTY` option.

2.2.3. DHCPv6 Client Behavior for IA_NA allocation

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

- o In the SOLICIT message within the IA_NA option, it **MAY** include the `OPTION_PREFIX_CLASS` requesting address to be allocated from a specific class indicated in that option. The class information to be requested can be learnt via configuration or any other out of band mechanism not described in this document.

If DHCPv6 client receives `OPTION_PREFIX_CLASS`, `OPTION_PREFIX_PROPERTY` options in the IAaddr-options area of the corresponding IA_NA but does not support one or both of these options, it **SHOULD** ignore the received option(s).

2.2.4. DHCPv6 Server Behavior for IA_NA allocation

The DHCPv6 server parses `OPTION_PREFIX_CLASS` option received and when it supports allocation within the requested `OPTION_PREFIX_CLASS` responds with an ADVERTISE message after populating the IA_NA option with address information from the requested prefix class. Along with including the IA Address options in the IA_NA option, it also includes the `OPTION_PREFIX_CLASS` option in the corresponding IAaddr-options area.

Even though the IA_NA option in the SOLICIT message does not include the `OPTION_PREFIX_CLASS` option, based on local policies, the DHCP server **MAY** select a default `OPTION_PREFIX_CLASS` value for the client and then **SHOULD** include the `OPTION_PREFIX_CLASS` option in the IAaddr-options area of the corresponding IA_NA it sends to the client. If both DHCP client and server support class based address allocation, but the DHCP server cannot offer addresses in the specified Usage class then the DHCP server **MUST** include the status code NoAddrsAvail (as defined in [RFC3315]) in the response message. If the DHCP server cannot offer addresses for any other reason, it **MUST** respond to client with appropriate status code as specified in [RFC3315]. In

addition if the server has additional property associated with the prefix by means of configuration or learnt from DHCPv6 prefix delegation or derived via any other means it MUST be sent as OPTION_PREFIX_PROPERTY option.

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. Prefix property and class can be used for source address selection by applications using the prefix for communication.

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" (1), "guest"(2), "voice" (3) etc), for assigning the IPv6 addresses to the end hosts through DHCPv6 IA_NA based on a preconfigured mapping with OPTION_PREFIX_CLASS option, the following conditions MAY be observed:

- o It MAY have a pre-configured mapping between the prefix class and OPTION_USER_CLASS option received in IA_NA.
- o It MAY match the OPTION_PREFIX_CLASS if the IA_NA request received contains OPTION_PREFIX_CLASS.
- o It MAY have a pre-configured mapping between the prefix class and the client DUID received in DHCPv6 message.
- o It MAY have a pre-configured mapping between the prefix class and its network interface on which the IA_NA request was received.

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

2.3.2. Class based prefix and IA_PD allocation

If the requesting router, receives prefix(es) for different classes (for example "video"(1), "guest"(2), "voice"(3) 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, it is specified in [I-D.korhonen-6man-prefix-properties].

2.3.4. Class based prefix and applications

Applications within a host can do source address selection based on the class of the prefix learnt in OPTION_PREFIX_PROPERTY and OPTION_PREFIX_CLASS using rules defined in [RFC6724]. The internal data structure and interface for source address selection used by application to choose source prefix with specific property and class in a host is beyond the scope of this document.

3. Example Application

3.1. Mobile gateway example

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.

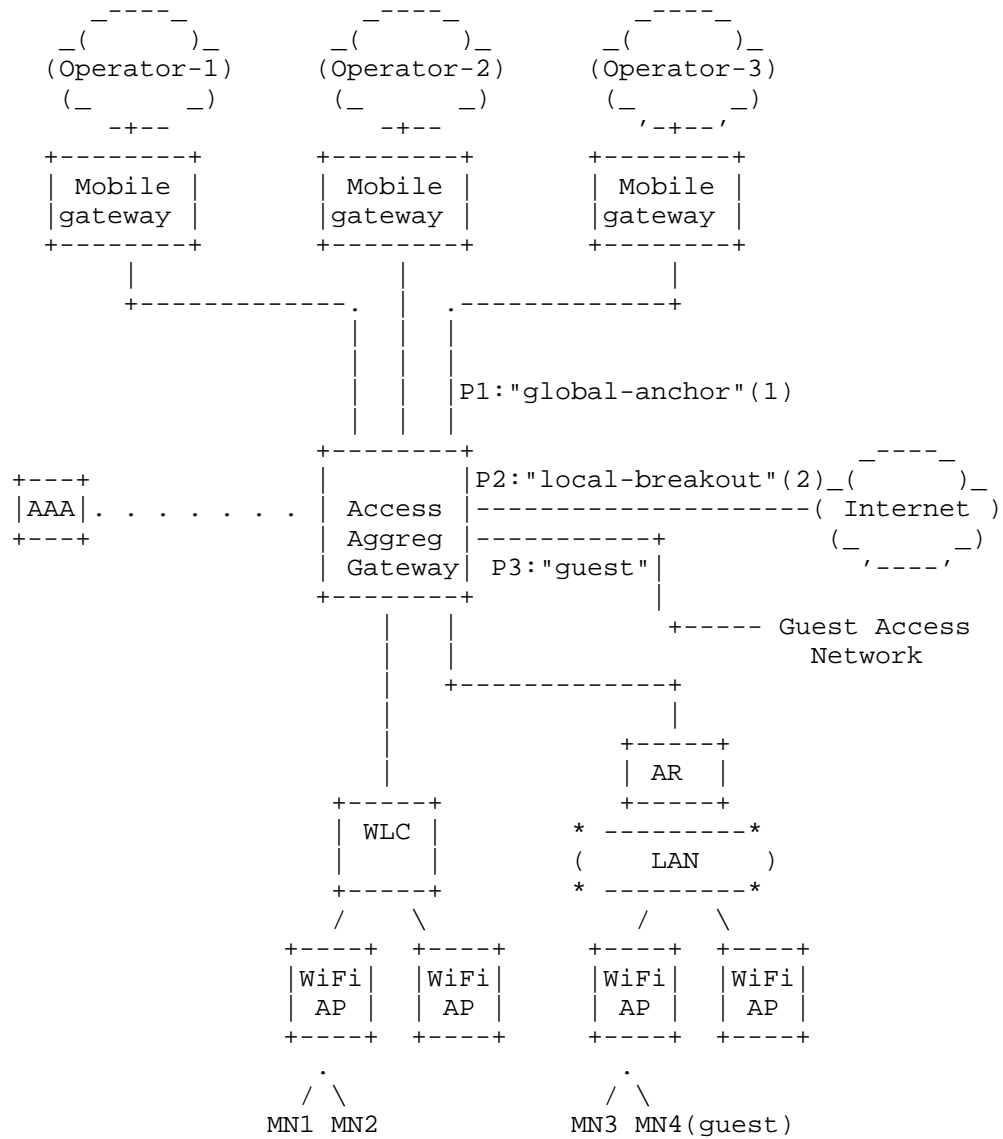


Figure 1: Example mobile network

3.1.1. Class based prefix delegation

The Access Aggregation Gateway requests for Prefix delegation from Mobile gateway and associates the prefix received with class "global-anchor"(1). The Access Aggregation Gateway is preconfigured to provide prefixes from the following classes: "global-anchor" (1), "local-breakout"(2), "guest"(3). 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"(1)
2. P2: IA_PD Prefix option with a prefix 3001:2::/64 containing OPTION_PREFIX_CLASS set to "local-breakout"(2)
3. P3: IA_PD Prefix option with a prefix 3001:3::/64 containing OPTION_PREFIX_CLASS set to "guest"(3)

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

3.1.2. 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"(1). 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"(3) 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

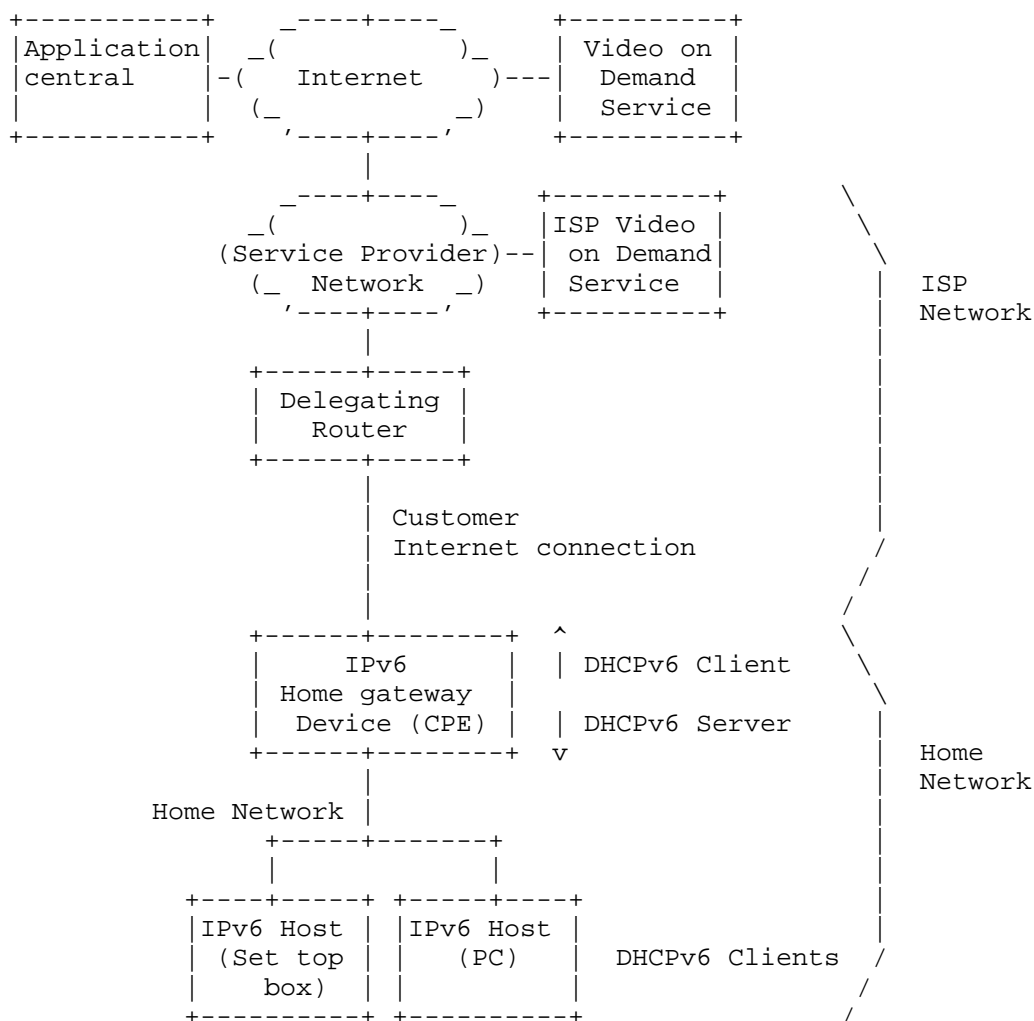
profile to provide both local-breakout Internet access and global-anchor, it offers an IPv6 address from the class "local-breakout" (2) and "global-anchor"(1). For MN1 it advertises 3001:2::1 and 3001:1::2 as the IPv6 address in OPTION_IAADDR in response to the IA_NA request. Applications within MN1 can choose to use the appropriate prefix based on the mobility enabled or local-breakout property attached to the prefix based on source address selection policy.

The prefixes that are globally anchored and hence have mobility can be advertised with OPTION_PREFIX_PROPERTY set to 0x0002 to convey that the prefix provides network based mobility as listed in Section 6.1. If the prefix also provides security guarantees OPTION_PREFIX_PROPERTY can be set to 0x000A to indicate mobility and security guarantees by bitwise ORing of both the properties.

3.2. Homenet Example

The following sub-section describes an example of class based prefix delegation in a home network environment. The network consists of the following elements:

- o Home Gateway (HGW) device: a routing device located in the customer's premises that provides connectivity between the customer and the service provider. In this example, the HGW is functioning as both a DHCP client towards the service provider's DHCP infrastructure and a DHCP server towards hosts located in the home network.
- o IPv6 Set Top Box (STB): A dedicated, IPv6 attached, video on demand device.
- o IPv6 PC: An IPv6 attached personal computer
- o Delegating Router: The router in the ISPs network acting as a DHCP server for the IA_PD request.
- o ISP Video On Demand (ISP-VOD) service: An ISP provided service offering unicast based streaming video content to subscribers.
- o Video On Demand (VOD) service: A server providing unicast based streaming video content to subscribers
- o On demand Video Application: Application hosted on the IPv6 PC
- o Application Central: Application server hosted in the Internet that the On demand Video Application communicates with to access VOD service



Simple home network with Data and Video devices

3.2.1. Class based prefix delegation to the HGW

In this example, three different services are being run on the same network. The service provider wishes that traffic is sourced from different prefixes by the home network clients [I-D.jiang-v6ops-semantic-prefix]. The HGW (requesting router) has been configured to request prefix delegation from the ISPs delegating router with the usage classes "video" (1) and "internet"(2) and "video-app" (3) the meaning of these being of relevance to the ISP

operating this and application that are configured out of band to utilize it.

The delegating router is preconfigured to advertise prefixes with these service classes. The HGW sends three IA_PD options within the SOLICIT message, one with OPTION_PREFIX_CLASS "video" (1), the second with "internet" (2) and a third with "video-app" (3). The HGW receives an advertise with the following prefixes in the IA_PD option:

1. P1: IA_PD Prefix option with a prefix 3001:5::/56 containing OPTION_PREFIX_CLASS set to "video" (1) with OPTION_PREFIX_PROPERTY set to 0x0001 indicating there is no internet reach
2. P2: IP_PD Prefix option with a prefix 3001:6::/56 containing OPTION_PREFIX_CLASS set to "internet" (2)
3. P3: IP_PD Prefix option with a prefix 3001:7::/56 containing OPTION_PREFIX_CLASS set to "video-app" (3) with property set to 0x0040 indicating the prefix provides Internet service SLA

It sends a REQUEST message with all of the above prefixes and receives a REPLY message with prefixes allocated for each of the requested classes. The HGW then configures a /64 prefix from each of the delegated prefixes on its LAN interface [RFC6204] and sends out router advertisements (RAs) with the "M" and "O" bits set.

3.2.2. IPv6 Assignment to Homenet hosts using stateful DHCPv6

1. STB sends a DHCPv6 SOLICIT message with the OPTION_PREFIX_CLASS option set to "video" (1) within the IA_NA. The HGW checks the requested prefix class against the available prefixes it has been delegated and advertises 3001:5::1 to the STB. The STB then configures this address on its LAN interface and uses it for sourcing requests to the VOD service.
2. The PC sends a DHCPv6 SOLICIT message requesting for IA_NA with the OPTION_PREFIX_CLASS option in ORO indicating support for prefix class. The HGW checks the available prefixes it has been delegated and advertises IA_NA from P1 (3001:5:2 with property set to 0x0001) , P2 (3001:6::1) and P3 (3001:7::1) to the PC or in absence of OPTION_PREFIX_CLASS in the solicit HGW is preconfigured to assign from the "internet"(2) class as the default. The PC then configures these addresses on its LAN interface and uses it for sourcing requests to the Internet.
3. The On demand Video Application on the PC communicates with its well known Application Central using the "internet" prefix and is

directed to use "video-app" prefix if available based on agreement between service provider and on demand video application service provider. The On demand Video Application then connects using the address assigned from P3 that will offer better experience based on the SLA between the providers.

4. If the homenet hosts use SLAAC prefix delegation with the class will use the options and procedure defined in [I-D.korhonen-6man-prefix-properties]

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, Maciek Konstantynowicz

5. Contributors

Authors would like to thank contributions to use cases and text for various sections received from Sindhura Bandi.

6. IANA Considerations

IANA is requested to assign an option code to OPTION_PREFIX_PROPERTY (TBD1) and OPTION_PREFIX_CLASS (TBD2) from the "DHCPv6 and DHCPv6 options" registry (<http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xml>).

6.1. OPTION_PREFIX_PROPERTY values

IANA is requested to reserve and maintain registry of OPTION_PREFIX_PROPERTY values and manage allocation of values as per as per policy defined in [RFC5226] with IETF assigned values requiring IETF consensus, RFC Required policy, any other values other than the ones listed below are not valid.

1. 0x0001 The prefix cannot be used to reach the Internet
2. 0x0002 The prefix provides network based mobility
3. 0x0004 The prefix requires authentication
4. 0x0008 The prefix is assigned on an interface that provides security guarantees
5. 0x0010 Usage is charged

6. 0x0020 The prefix provides multi-homed redundancy
 7. 0x0040 The prefix provides Internet service SLA, based on associated OPTION_PREFIX_CLASS
 8. 0x0080 Unassigned
 9. 0x0100 Unassigned
 10. 0x0200 Unassigned
 11. 0x0400 Unassigned
 12. 0x0800 Unassigned
 13. 0x1000 Unassigned
 14. 0x2000 Unassigned
 15. 0x4000 Unassigned
 16. 0x8000 Unassigned
7. 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.
8. 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
- Changes from -01 to -02
- a. Modified option format to be enumerated values
 - b. Added IANA section to request managing of registry for the enumerated values
 - c. Added initial values for the class

- d. Added section for applications to select address with a specific property

Changes from -02 to -03

- a. Added server behaviour for IA_NA and IA_PD allocation
- b. Added Class based Information-Request usage

Changes from -03 to -04

- a. Added homenet use case
- b. Split usage class into a fixed IANA maintained properties registry and a prefix class

Changes from -04 to -05

- a. Added on demand video application use case and modified the example section
- b. Added additional properties and reference for SLAAC/ND procedure

9. References

9.1. Normative References

- [I-D.ietf-dhc-dhcpv4-over-ipv6]
Cui, Y., Wu, P., Wu, J., and T. Lemon, "DHCPv4 over IPv6 Transport", draft-ietf-dhc-dhcpv4-over-ipv6-06 (work in progress), March 2013.
- [I-D.jiang-v6ops-semantic-prefix]
Jiang, S., Sun, Q., Farrer, I., and Y. Bo, "A Framework for Semantic IPv6 Prefix", draft-jiang-v6ops-semantic-prefix-03 (work in progress), May 2013.
- [I-D.korhonen-6man-prefix-properties]
Korhonen, J., Patil, B., Gundavelli, S., Seite, P., and D. Liu, "IPv6 Prefix Properties", draft-korhonen-6man-prefix-properties-02 (work in progress), July 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC6204] Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", RFC 6204, April 2011.
- [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

9.2. Informative References

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

Authors' Addresses

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

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

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

Laurent Thiebaut
Alcatel-Lucent
France

Email: laurent.thiebaut@alcatel-lucent.com

Jouni Korhonen
Renesas Mobile
Linnoitustie 6
FIN-02600 Espoo
Finland

Email: jouni.nospam@gmail.com

Ian Farrer
Deutsche Telekom AG
GTN-FM4, Landgrabenweg 151
Bonn 53227
Bonn 53227

Email: ian.farrer@telekom.de

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 10, 2013

Z. Cao
T. Sun
China Mobile
S. McCann
Research in Motion
July 9, 2012

DHCPv4 and DHCPv6 Options for Access Network Query Protocol Servers
draft-cao-dhc-anqp-option-00

Abstract

This document defines a DHCPv4 option and DHCPv6 option of the Access Network Query Protocol (ANQP) server address. These options are used to configure the ANQP server addresses on the Access Point of WLAN system.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 10, 2013.

Copyright Notice

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

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

described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Language	3
2. Terminology	3
3. DHCPv4 Option of ANQP Server Address	4
4. DHCPv6 Option of ANQP Server Address	5
5. Advertisement Server Type	6
6. IANA Considerations	6
7. Security Considerations	7
8. References	7
8.1. Normative References	7
8.2. Informative References	7
Authors' Addresses	7

1. Introduction

Access Network Query Protocol (ANQP) was defined by IEEE 802.11u Task Group [IEEE-ELEVENU] and is now integrated into the 802.11-2012 specification suite. And ANQP has been further extended by the Hotspot 2.0 Technical Group of Wi-Fi Alliance (WFA), and it has been included in the representative certification program called "Passpoint" [PASSPOINT].

ANQP is an example of the query protocol for access network information retrieval, and it is transported by the IEEE 802.11 defined Generic Advertisement Service (GAS) Public Action frames. GAS enables a WLAN client (e.g., a STA) to exchange messages with an advertisement server (e.g., an ANQP server) in the pre-association state, i.e., prior to association. With the information retrieved via this server, the WLAN client connection manager can make informed selection among multiple access networks. One example of using ANQP is that the WLAN client in a roaming environment can select the correct visited access network that has roaming relationship with its home service provider without user intervention.

In a scalable deployment environment, the ANQP server will not be placed on the Access Point (AP), rather it should be placed on a centralized device that serves different APs. The AP will forward the ANQP message on the IP network between AP and ANQP Server. Then the problem of configuring the ANQP server address on the AP arises.

This document defines a DHCPv4 option and DHCPv6 option of the ANQP server addresses. As introduced above, these options are used to configure the ANQP server addresses on the APs. This document also defines the "Advertisement Protocol Type" field in the DHCPv4/v6 options which can be extended to configure other types of advertisement protocols servers.

1.1. Requirements Language

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

2. Terminology

ANQP: Access Network Query Protocol. ANQP is an example of a query protocol for access network information retrieval transported by Generic Advertisement Service (GAS) Public Action frames defined in IEEE 802.11. ANQP message exchanges happen before network association. ANQP is defined in the IEEE 802.11 specification and

has been further extended by the Wi-Fi Alliance.

ANQP Server: ANQP Server is the network entity that terminates and responds to ANQP enquiries. In a scalable deployment, the ANQP Server is placed in centralized device and administrated by the Wi-Fi server provider.

IEEE 802.11u: IEEE 802.11u-2011 is an amendment to the IEEE 802.11-2007 standard that added features that improve interworking with external networks. It is now incorporated within IEEE 802.11-2012. A key amendment to IEEE 802.11-2012 is the capability of WLAN client network discovery and selection.

Passpoint: Wi-Fi Alliance Certified Program Name. The technical specification of Passpoint is based on the output of the WFA Hotspot 2.0 (HS2.0) Technical Task Group. HS2.0 defines further vendor specific ANQP options and has developed a test plan for Passpoint certification.

RLQP: Registered Location Query Protocol. This is an additional advertisement protocol defined by IEEE 802.11af [RLQP] (TV White Spaces), which assists with location information, but operates as a separate RLQP Server. The RLQP Server and ANQP Server may be co-located.

3. DHCPv4 Option of ANQP Server Address

This section describes the ANQP Server Address Option for DHCPv4. The option layout is depicted below Figure 1:

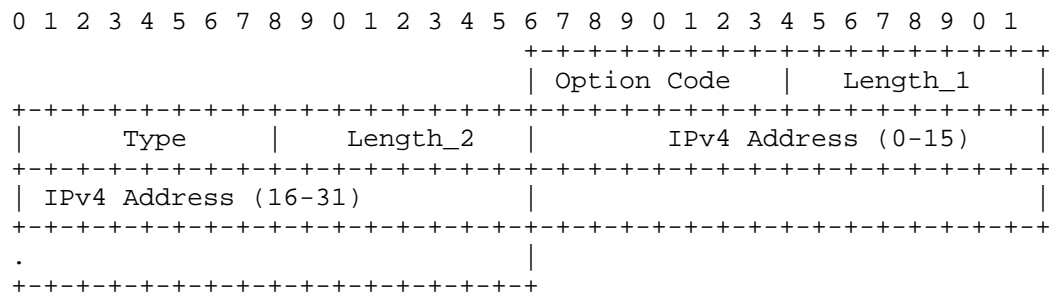


Figure 1: ANQP Server Address Option for DHCPv4

Option Code: OPTION-IPv4_Address-Adv-Server

Length_1 Length (in bytes) of the option excluding the 'Option Code' and the 'Length_1' fields;

Type (Advertisement Server Type): Indicates the type of the advertisement server. There are different advertisement servers defined in 802.11, including ANQP and RLQP. The values of those server types are discussed in Section 5.

Length_2: Length (in bytes) of the IPv4 addresses of the advertisement server; its value equals four times of the number of IPv4 addresses (4*N);

IP Address: IPv4 address(es) of ANQP server(s)

4. DHCPv6 Option of ANQP Server Address

This section describes the ANQP Server Address Option for DHCPv6. The option layout is depicted below Figure 2:

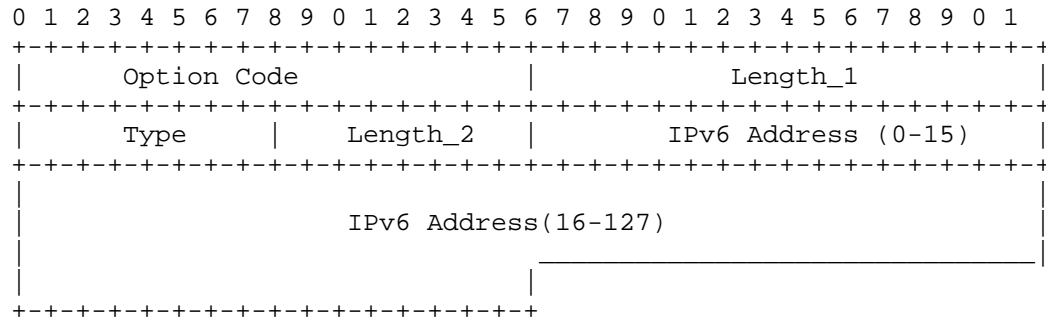


Figure 2: ANQP Server Address Option for DHCPv6

Option Code: OPTION-IPv6_Address-Adv-Server

Length_1: Length (in bytes) of the option excluding the 'Option Code' and the 'Length_1' fields;

Type (Advertisement Server Type): Indicates the type of the advertisement server. There are different advertisement servers defined in 802.11, including ANQP and RLQP. The values of those server types are discussed in Section 5.

Length_2: Length (in bytes) of the IPv4 addresses of the advertisement server; its value equals 16 times of the number of IPv6 addresses (16*N);

IP address: IPv6 address(es) of ANQP server(s)

5. Advertisement Server Type

There are different types of advertisement servers defined in 802.11, including ANQP Server and RLQP Server. IEEE may define other advertisement servers in future. To make options defined in this document scalable to further extensions, and also avoid the need of an individual option code for each of such advertisement servers, this document defines the Advertisement Server Type field in both the DHCPv4 and DHCPv6 options.

The Advertisement Server Type value of ANQP is suggested in this document as below.

Type	Value
Reserved	0
ANQP	1
Reserved	2-255

In addition to ANQP, other advertisement protocols have been defined within IEEE 802.11 (e.g. RLQP). These operate in a similar manner to ANQP, but allow information exchange with different servers than that of the ANQP Server. The Advertisement Server Type value of other protocols including RLQP will be extended by future work.

6. IANA Considerations

This document has the following requests to the IANA.

Option Code for OPTION-IPv4_Address-Adv-Server in DHCPv4, as defined in Section. 3 of this document.

Option Code for OPTION-IPv6_Address-Adv-Server in DHCPv6, as defined in Section. 4 of this document.

Advertisement Server Type for ANQP, as defined in Section. 5 of this document.

7. Security Considerations

If adversaries are able to forge rogue ANQP Server options, the ANQP messages will be directed to wrong servers and bogus information about the queried access network would be injected. The DHCP authentication option described in [RFC3315] and [RFC3118] MAY be used to mitigate the above attacks. Lower layer security such as L2 traffic filtering and firewall SHOULD be configured to prevent such attacks.

8. References

8.1. Normative References

- [IEEE-ELEVENU]
IEEE, "IEEE 802.11u Specification", 2011, <<http://standards.ieee.org/findstds/standard/802.11-2012.html>>.
- [PASSPOINT]
Wi-Fi Alliance, "Wi-Fi CERTIFIED Passpoint", 2012, <<http://www.wi-fi.org/discover-and-learn/wi-fi-certified-passpoint>>.
- [RLQP]
"Wireless LAN in the TV White Space", 2012, <http://www.ieee802.org/11/Reports/tgaf_update.htm>.

8.2. Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

Authors' Addresses

Zhen Cao
China Mobile
Xuanwumenxi Ave. No. 32
Beijing, 100871
China

Phone: +86-10-52686688
Email: zehn.cao@gmail.com, caozhen@chinamobile.com

Tao Sun
China Mobile
Xuanwumenxi Ave. No. 32
Beijing, 100871
China

Phone: +86-10-52686688
Email: suntao@chinamobile.com

Stephen McCann
Research in Motion
200 Bath Road
Slough, SL1 3XE,
United Kingdom

Phone: +44 1754 66700
Fax:
Email: smccann@rim.com
URI:

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 12, 2013

G. Halwasia
S. Bhandari
W. Dec
Cisco Systems
March 11, 2013

Client Link-layer Address Option in DHCPv6
draft-ietf-dhc-dhcpv6-client-link-layer-addr-opt-05

Abstract

This document specifies the format and mechanism that is to be used for encoding client link-layer address in DHCPv6 Relay-Forward 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 12, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Problem Background and Scenario	2
3. DHCPv6 Client Link-layer Address Option	3
4. DHCPv6 Relay Agent Behavior	4
5. DHCPv6 Server Behavior	4
6. DHCPv6 Client Behavior	5
7. IANA Considerations	5
8. Security Considerations	5
9. Acknowledgements	6
10. References	6
10.1. Normative References	6
10.2. Informative References	6
Authors' Addresses	7

1. Introduction

This specification defines an optional mechanism and the related DHCPv6 option to allow first-hop DHCPv6 relay agents (relay agents that are connected to the same link as the client) to provide the client's 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 link-layer address in the DHCPv4 message header. DHCPv4 message header has 'htype' and 'chaddr' fields to specify client link-layer address type and link-layer address respectively. The client link-layer 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, which results in a dual-stack network environment, there will be devices that act as both DHCPv4 and DHCPv6 clients. In service provider deployments, a typical DHCPv4 implementation will use the client link-layer address as one of the keys to build DHCP client lease database. In dual stack scenarios operators need to be able to associate DHCPv4 and DHCPv6 messages with the same client interface,

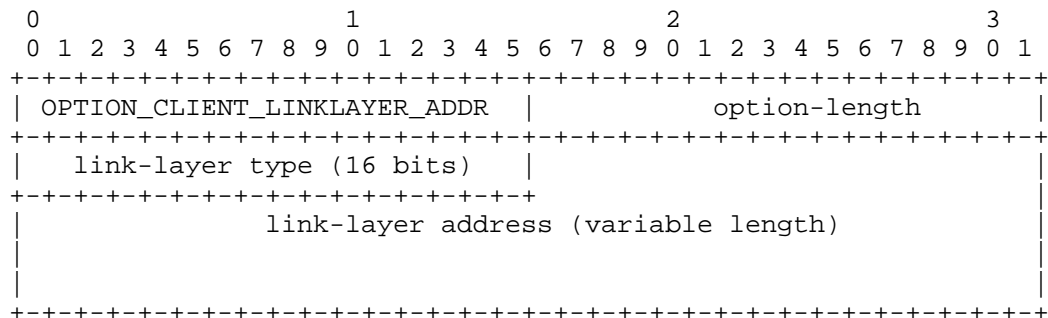
based on an identifier that is common to the interface. The client link-layer address is such an identifier.

Currently, the DHCPv6 protocol specification [RFC3315] does not define a way to communicate the client link-layer address to the DHCP server in cases where the DHCP server is not connected to the same network link as the DHCP client. 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 link-layer address as the customer identifier, and desires to correlate DHCPv6 assignments using the same identifier. [RFC4361] describes a mechanism for using the same DUID in both DHCPv4 and DHCPv6. Unfortunately, this specification requires modification of existing DHCPv4 clients, and has not seen broad adoption in the industry (indeed, we are not aware of any commercial implementations).

Providing an option in DHCPv6 Relay-Forward messages to carry client link-layer address explicitly will help above mentioned scenarios. For example, 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.



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

4. DHCPv6 Relay Agent Behavior

DHCPv6 Relay agents which receive messages originating from clients (for example Solicit and Request, but not, for example, Relay-Forward or Advertise) MAY include the link-layer source address of the received DHCPv6 message in Client Link-layer Address option in relayed DHCPv6 Relay-Forward messages. The DHCPv6 Relay agent behavior can depend on configuration that decides whether the Client Link-layer Address option needs to be included.

5. DHCPv6 Server Behavior

If DHCPv6 Server is configured to store or use client link-layer address, it SHOULD look for the client link-layer address option in the Relay-Forward DHCP message of the DHCPv6 Relay agent closest to the client. The mechanism described in this document is not necessary in the case where the DHCPv6 Server is connected to the same network link as the client, because the server can obtain the link-layer address from the link-layer header of the DHCPv6 message. If the DHCP server receives a Client Link-layer Address option anywhere in any encapsulated message that is not a Relay-Forward DHCP message, the server MUST silently ignore that option.

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

6. DHCPv6 Client Behavior

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

7. IANA Considerations

IANA is requested to assign an option code to OPTION_CLIENT_LINKLAYER_ADDR from the "DHCP Option Codes" registry (<http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xml>).

8. Security Considerations

It is possible for a rogue DHCPv6 relay agent to insert an incorrect Client Link Layer Address option for malicious purposes. A DHCPv6 client can also pose as a rogue DHCP relay agent, sending a Relay-Forward message containing an incorrect Client Link Layer Address option. In either case, it would be possible for a DHCPv6 client to masquerade as the same device as a DHCPv4 client, when in fact the two are distinct.

One possible attack that could be accomplished using this masquerade would be in the case where a DHCPv4 client is using DHCPv4 to do a Dynamic DNS update to install an A record so that it can be reached by other nodes [RFC4702]. A masquerading DHCPv6 client could use DHCPv6 to install an AAAA record with the same name [RFC4704]. Dual-stack nodes attempting to connect to the DHCPv4 client might then be tricked into connecting to the masquerading DHCPv6 client instead.

It is possible that there are other attacks that could be accomplished using this masquerading technique, although the authors are not aware of any. To prevent masquerades of this sort, DHCP server administrators are strongly advised to configure DHCP servers that use this option to communicate with their relay agents using IPsec as described in Section 21.1 of [RFC3315].

In some networks, it may be the case that the operator of the physical network and the provider of connectivity over that network are administratively separate, such that the client link-layer address option would reveal information to one or the other party that they do not need and could not otherwise obtain. It is also possible in some cases that a relay agent might communicate with a DHCP server over an open network where eavesdropping would be possible. In these cases, it is strongly recommended, in order to

protect end-user privacy, that network operators use IPsec to provide confidentiality for messages between the relay agent and DHCP server.

9. Acknowledgements

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

10. References

10.1. Normative References

- [RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC4361] Lemon, T. and B. Sommerfeld, "Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)", RFC 4361, February 2006.

10.2. Informative References

- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC4702] Stapp, M., Volz, B., and Y. Rekhter, "The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option", RFC 4702, October 2006.
- [RFC4704] Volz, B., "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option", RFC 4704, October 2006.

Authors' Addresses

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

Phone: +91 80 4429 2703
Email: ghalwasi@cisco.com

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

Phone: +91 80 4429 2627
Email: shwethab@cisco.com

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

Email: wdec@cisco.com

DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 30, 2014

L. Yeh
Freelancer Technologies
M. Boucadair
France Telecom
July 29, 2013

RADIUS Option for DHCPv6 Relay Agent
draft-ietf-dhc-dhcpv6-radius-opt-14

Abstract

The DHCPv6 RADIUS option provides a mechanism to exchange authorization and identification information between DHCPv6 relay agent and DHCPv6 server. This architecture assumes that the Network Access Server(NAS) acts as both DHCPv6 relay agent and RADIUS client. When receiving messages from the DHCPv6 clients, the NAS consults the RADIUS server and adds the RADIUS response when forwarding the DHCPv6 client's messages to the DHCPv6 server. The DHCPv6 server then uses that additional information to generate appropriate response to the DHCPv6 client's requests.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 30, 2014.

Copyright Notice

Copyright (c) 2013 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. DHCPv6 RADIUS option	7
4.1. RADIUS attributes permitted in DHCPv6 RADIUS option	8
5. DHCPv6 Relay Agent Behavior	8
6. DHCPv6 Server Behavior	8
7. DHCPv6 Client Behavior	8
8. Security Considerations	9
9. IANA Considerations	9
10. Acknowledgements	10
11. References	10
11.1. Normative References	10
11.2. Informative References	11
Authors' Addresses	11

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] and IPv6 prefix [RFC3633]. The stateless configuration parameters [RFC3736] include, for example, DNS [RFC3646], or a FQDN of AFTR [RFC6334]. In the scenarios described in this document, the DHCPv6 server is deployed in the central part of an ISP network.

RADIUS [RFC2865] is widely used as the centralized authentication, authorization and user management mechanism for service provision in Broadband access network. [RFC3162], [RFC4818], [RFC6519] and [RFC6911] specified the attributes that support the service provision for IPv6-only and IPv6-transition access. The RADIUS server authorizes the Network Access Server (NAS) to assign an IPv6 address or prefix from the indicated pool, or to assign an IPv6 address or prefix with an explicitly indicated value, and other configuration parameters as per the attributes for the subscribers.

When the NAS acts as distributed DHCPv6 server and RADIUS client simultaneously, it communicates with RADIUS server after receiving request from DHCPv6 client. Upon receiving the Access-Accept message from the RADIUS server, the NAS then responds to the DHCPv6 client's requests per the associated authorization information indicated by the RADIUS attributes in the Access-Accept message. When NAS acts as DHCPv6 relay agent and RADIUS client simultaneously, and the centralized DHCPv6 server is co-located with the RADIUS server, they may share the same database of the users; but when the centralized DHCPv6 server is not located in the same place as the RADIUS server, a new communication mechanism is needed for the DHCPv6 relay agent to transfer the authorization information indicated by the RADIUS attributes to the DHCPv6 server.

2. Terminology and Language

This document specifies a new DHCPv6 option for the DHCPv6 Relay Agent to transfer the authorization information of RADIUS attributes received in the Access-Accept message from the RADIUS server to the centralized DHCPv6 server. Definitions for terms and acronyms not specified in this document are defined in [RFC2865] and [RFC3315].

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. Network Scenarios

Figure 1 and Figure 2 show 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 DHCPv6 relay agent and the RADIUS client simultaneously. Figure 1 shows the sequence of DHCPv6 and RADIUS messages for IP over Ethernet (IPoE) access model, when the access loop adopts the direct Ethernet encapsulation. Figure 2 shows the sequence of DHCPv6 and RADIUS messages for PPP over Ethernet (PPPoE) access model.

The mechanism introduced in this document is a generic mechanism, and might also be employed in other network scenarios where the DHCPv6 relay agent and the RADIUS client locate in the same device.

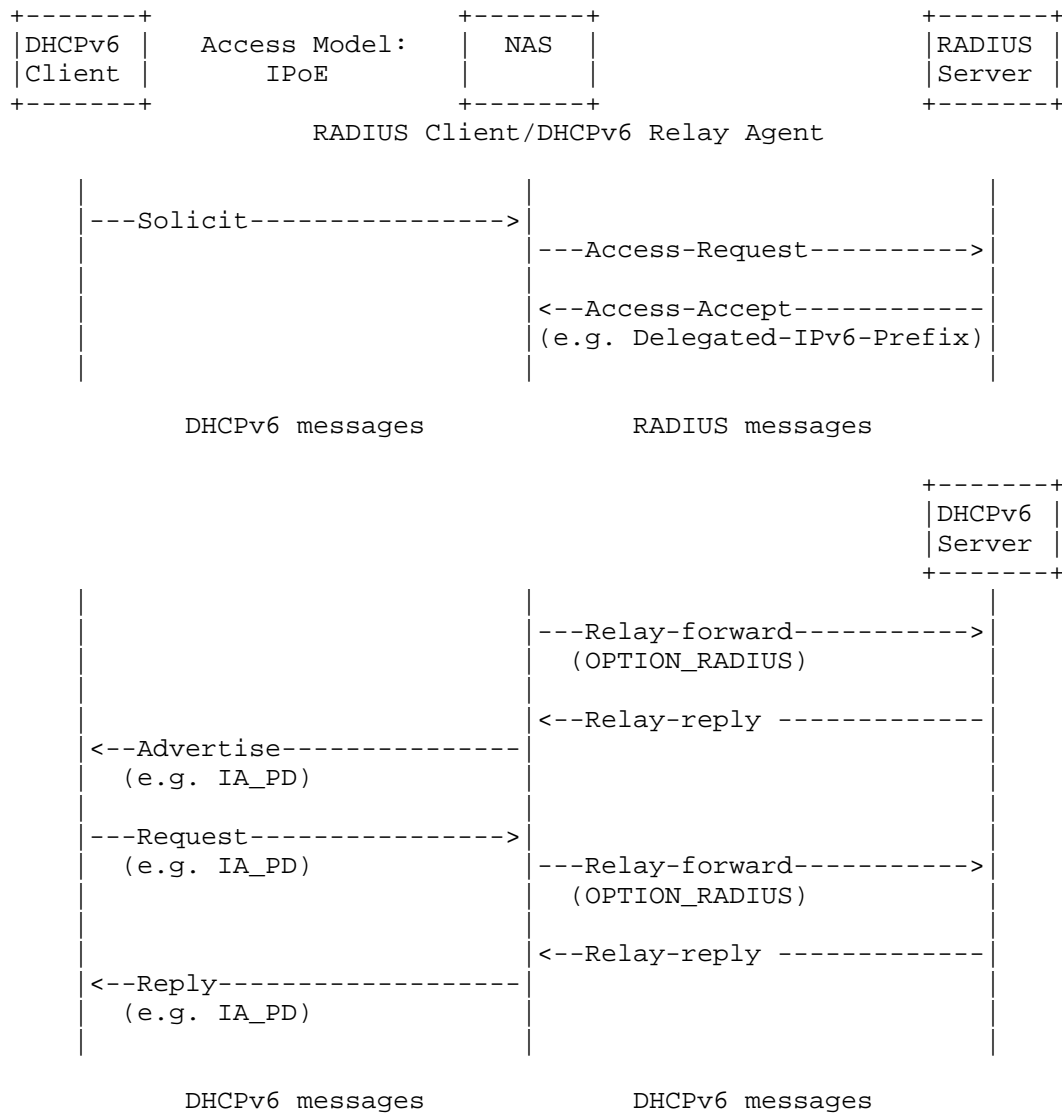


Figure 1: Network scenario and message sequence when employing DHCPv6 RADIUS option in IPoE access

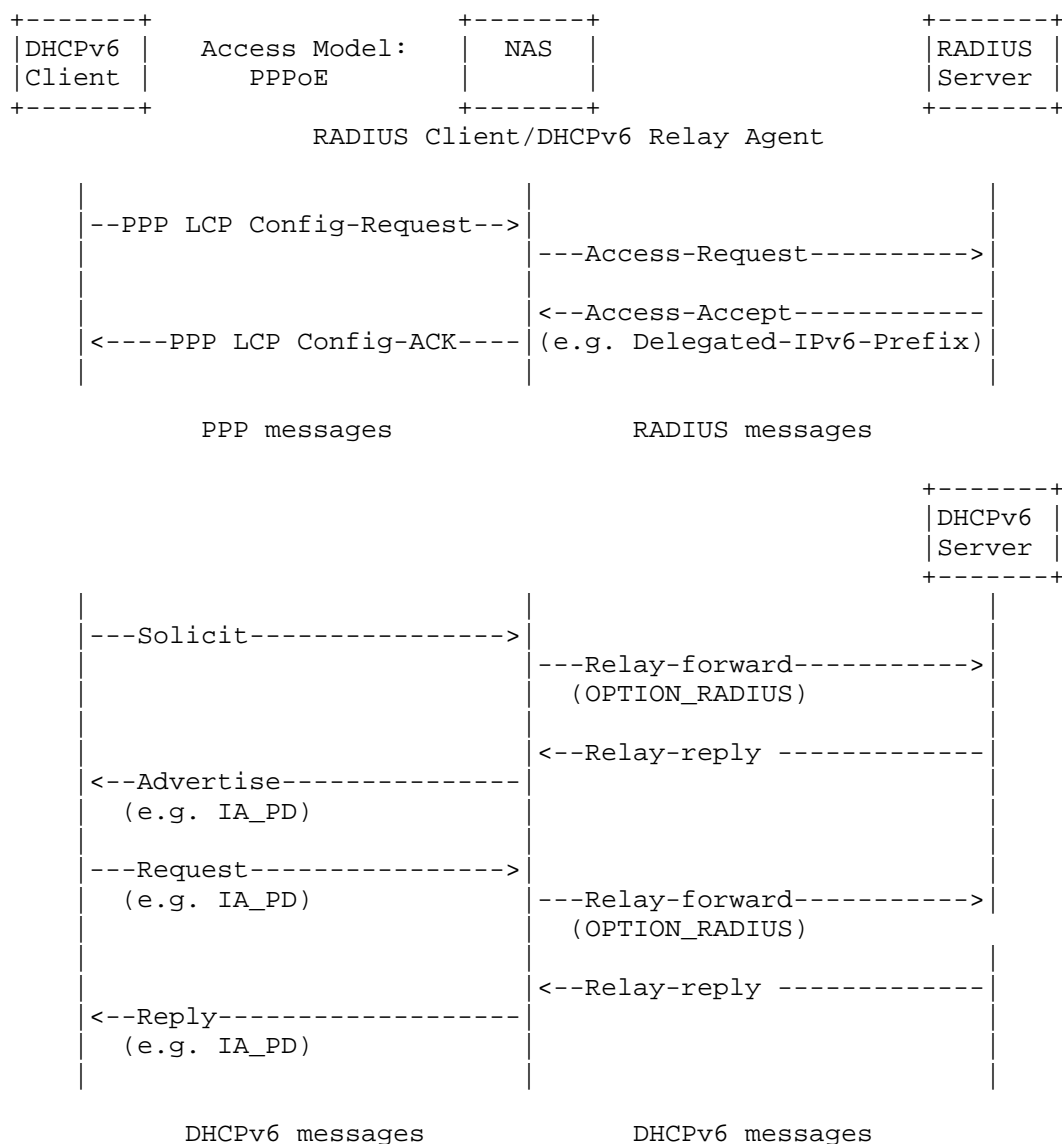


Figure 2: Network scenario and message sequence when employing DHCPv6 RADIUS option in PPPoE access

If the authentication or the authorization through RADIUS fails, the associated message sequences will stop. The NAS acting as the DHCPv6 relay agent will not forward the message received from the client to the DHCPv6 server. If the authentication or the authorization through RADIUS passes, the NAS MUST store the information indicated

in the RADIUS attributes received in the Access-Accept message from the RADIUS server during the whole session. How the NAS manages these information during the RADIUS session is out of the scope of this document.

After receiving RENEW (5) message from the DHCPv6 client, the NAS SHOULD NOT initiate a new Access-Request/Access-Accept message exchange with the RADIUS server. After receiving REBIND (6) message from the DHCPv6 client, the NAS MUST initiate a new Access-Request/Access-Accept message exchange with the RADIUS server, unless RADIUS capability is disabled on the NAS.

4. DHCPv6 RADIUS option

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

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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|               OPTION_RADIUS               | option-len |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               option-data (List of RADIUS Attributes)
+-----+-----+-----+-----+-----+-----+-----+-----+

```

option-code	TBD
option-len	Length of the option-data in octets
option-data	List of one or more RADIUS attributes

The option-data of OPTION_RADIUS is a list of one or more RADIUS attributes received in the Access-Accept message from the RADIUS server. The format of RADIUS attributes is defined in section 5 of [RFC2865] as well as sections 2.1 and 2.2 of [RFC6929]. If multiple attributes with the same type (including the Long Extended type defined in sections 2.2 of [RFC6929]) are present, the order of attributes with the same type MUST be the same as that received from the RADIUS server. The OPTION_RADIUS can only contain the RADIUS attributes listed in the IANA Registry of 'RADIUS attributes permitted in DHCPv6 RADIUS option'.

According to the network scenarios described in section 3, the OPTION_RADIUS should appear in the RELAY-FORW (12) message relaying SOLICIT (1), REQUEST (3) and REBIND (6) from the DHCPv6 client, and may appear in the RELAY-FORW (12) relaying any other message from the

DHCPv6 client.

4.1. RADIUS attributes permitted in DHCPv6 RADIUS option

The RADIUS attributes listed in the below table are recommended as the first batch of attributes in the IANA Registry of 'RADIUS attributes permitted in DHCPv6 RADIUS option'. New RADIUS attributes can be added to this list after Expert Review [RFC5226].

Type Code	Attribute	Reference
26	Vendor-Specific	[RFC2865]
123	Delegated-IPv6-Prefix	[RFC4818]
144	DS-Lite-Tunnel-Name	[RFC6519]
168	Framed-IPv6-Address	[RFC6911]
169	DNS-Server-IPv6-Address	[RFC6911]
171	Delegated-IPv6-Prefix-Pool	[RFC6911]
172	Stateful-IPv6-Address-Pool	[RFC6911]

Note: The RADIUS attribute's 'Length' defined in section 5 of [RFC2865] includes the length of 'Type' and 'Length' fields.

5. DHCPv6 Relay Agent Behavior

If the Relay Agent is configured to send OPTION_RADIUS, and the Access-Accept message from the RADIUS server contained RADIUS attributes permitted for use in OPTION_RADIUS, the Relay Agent MUST include OPTION_RADIUS in the RELAY-FORW (12) message. The DHCPv6 relay agent includes the permitted RADIUS attributes into OPTION_RADIUS one by one; if multiple attributes with the same type are present, the order of attributes with the same type MUST be the same as that received from the RADIUS server.

6. DHCPv6 Server Behavior

Upon receipt of the RELAY-FORW (12) message with OPTION_RADIUS from a relay agent, the DHCPv6 server that supports OPTION_RADIUS 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 MUST silently discard this option.

7. DHCPv6 Client Behavior

OPTION_RADIUS 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], section 11 of [RFC6929].

The mechanism described in this document may introduce new attack vector against the DHCPv6 server in case the DHCPv6 relay agent is compromised. By forging the RADIUS attributes contained in the OPTION_RADIUS of the RELAY-FORW (12) messages, the attacker may influence the parameter assignment on the DHCPv6 server for the DHCPv6 clients. However, as those network scenarios described in the section 3, NAS always belongs to the same administrative domain of the DHCPv6 server in the real deployment.

Network administrators should be aware that although RADIUS messages are encrypted, DHCPv6 messages are always not encrypted. It is possible that some RADIUS vendor-specific attributes might contain the sensitive or confidential information. Network administrators are strongly advised to prevent including such information into DHCPv6 messages.

If the use of vendor-specific attributes with confidential content is required, administrators are advised to use IPsec with encryption to protect the confidentiality of the RADIUS attributes. Relay agents and servers implementing this specification MUST support the use of IPsec ESP with encryption in transport mode according to section 3.1.1 of [RFC4303] and section 21.1 of [RFC3315].

9. IANA Considerations

This document requests to assign a new DHCPv6 option code for OPTION_RADIUS defined in section 4, and to create a new registry on the same assignment page, which is entitled as 'RADIUS attributes permitted in DHCPv6 RADIUS option' defined in section 4.1. The new registry will enumerate the RADIUS Attributes Types (<http://www.iana.org/assignments/radius-types/radius-types.xml>) that are permitted to be included in the DHCPv6 RADIUS option. The allocation policy of this 'RADIUS attributes permitted in DHCPv6 RADIUS option' registry is Expert Review [RFC5226]. Designated expert should carefully consider the security implications of

allowing the relay agent to include new RADIUS attribute for the addition to this registry.

10. Acknowledgements

Thanks to Tomek Mrugalski, Bernie Volz, Gaurav Halwasia and Roberta Maglione for their thorough review comments in the mailing list of DHC working group, to Ted Lemon for his continuous encouragement and technical guidance.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [RFC4818] Salowey, J. and R. Droms, "RADIUS Delegated-IPv6-Prefix Attribute", RFC 4818, April 2007.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC6519] Maglione, R. and A. Durand, "RADIUS Extensions for Dual-Stack Lite", RFC 6519, February 2012.
- [RFC6911] Dec, W., Sarikaya, B., Zorn, G., Miles, D., and B. Lourdelet, "RADIUS Attributes for IPv6 Access Networks", RFC 6911, April 2013.
- [RFC6929] DeKok, A. and A. Lior, "Remote Authentication Dial In User Service (RADIUS) Protocol Extensions", RFC 6929, April 2013.

11.2. Informative References

- [RFC3162] Aboba, B., Zorn, G., and D. Mitton, "RADIUS and IPv6", RFC 3162, August 2001.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.
- [RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.
- [RFC6334] Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite", RFC 6334, August 2011.

Authors' Addresses

Leaf Y. Yeh
Freelancer Technologies
P. R. China

Email: leaf.yeh.sdo@gmail.com

Mohamed Boucadair
France Telecom
France

Email: mohamed.boucadair@orange.com

Network Working Group
Internet-Draft
Updates: 3315,3633 (if approved)
Intended status: Standards Track
Expires: September 21, 2015

O. Troan
B. Volz
Cisco Systems, Inc.
M. Siodelski
ISC
March 20, 2015

Issues and Recommendations with Multiple Stateful DHCPv6 Options
draft-ietf-dhc-dhcpv6-stateful-issues-12.txt

Abstract

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) specification defined two stateful options, IA_NA and IA_TA, but did not anticipate the development of additional stateful options. DHCPv6 Prefix Delegation added the IA_PD option, which is stateful. Applications that use IA_NA and IA_PD together have revealed issues that need to be addressed. This document updates RFC 3315 and RFC 3633 to address these issues.

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 21, 2015.

Copyright Notice

Copyright (c) 2015 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	3
2. Conventions	3
3. Terminology	3
4. Handling of Multiple IA Option Types	4
4.1. Placement of Status Codes in an Advertise Message	5
4.2. Advertise Message Processing by a Client	7
4.3. T1/T2 Timers	8
4.4. Renew and Rebind Messages	9
4.4.1. Renew Message	9
4.4.2. Rebind Message	10
4.4.3. Updates to section 18.1.3 of RFC 3315	10
4.4.4. Updates to Section 18.1.4 of RFC 3315	12
4.4.5. Updates to Section 18.1.8 of RFC 3315	13
4.4.6. Updates to Section 18.2.3 of RFC 3315	15
4.4.7. Updates to Section 18.2.4 of RFC 3315	17
4.4.8. Updates to RFC 3633	18
4.5. Confirm Message	19
4.6. Decline Should Not Necessarily Trigger a Release	20
4.7. Multiple Provisioning Domains	21
5. IANA Considerations	21
6. Security Considerations	21
7. Acknowledgements	21
8. References	21
8.1. Normative References	21
8.2. Informative References	22
Authors' Addresses	22

1. Introduction

DHCPv6 [RFC3315] was written without the expectation that additional stateful DHCPv6 options would be developed. DHCPv6 Prefix Delegation [RFC3633] since added a new stateful option for Prefix Delegation to DHCPv6. Implementation experience of the Customer Edge Router (CER) model described in [RFC7084] has shown issues with the DHCPv6 protocol in supporting multiple stateful option types, in particular IA_NA (non-temporary addresses) and IA_PD (delegated prefixes).

This document describes a number of problems encountered with coexistence of the IA_NA and IA_PD option types and specifies changes to the DHCPv6 protocol to address these problems.

The intention of this work is to clarify and, where needed, modify the DHCPv6 protocol specification to support IA_NA and IA_PD option types within a single DHCPv6 session.

Note that while IA_TA (temporary addresses) options may be included with other IA option type requests, these generally are not renewed (there are no T1/T2 times) and have a separate life cycle from IA_NA and IA_PD option types. Therefore, the IA_TA option type is mostly out of scope for this document.

The changes described in this document are intended to be incorporated in a new revision of the DHCPv6 protocol specification ([I-D.dhcgw-dhc-rfc3315bis]).

2. Conventions

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. Terminology

In addition to the terminology defined in [RFC3315], [RFC3633], and [RFC7227], the following terminology is used in this document:

Identity association (IA): Throughout this document, "IA" is used to refer to the Identity Association containing addresses or prefixes assigned to a client and carried in the IA_NA or IA_PD options respectively.

IA option types: This is used to generally mean an IA_NA and/or IA_PD option.

Stateful options:	Options that require dynamic binding state per client on the server.
Top-level options:	Top-level options are DHCPv6 options that are not encapsulated within other options, excluding the Relay-Message option. Options encapsulated by Relay-message options, but not by any other option, are still top-level options, whether they appear in a relay agent message or a server message. See [RFC7227].

4. Handling of Multiple IA Option Types

The DHCPv6 specification [RFC3315] was written with the assumption that the only stateful options were for assigning addresses. DHCPv6 Prefix Delegation [RFC3633] describes how to extend the DHCPv6 protocol to handle prefix delegation, but does not clearly specify how the DHCP address assignment and prefix delegation co-exist.

If a client requests multiple IA option types, but the server is configured to only offer a subset of them, the client could react in several ways:

1. Reset the state machine and continue to send Solicit messages,
2. Create separate DHCP sessions for each IA option type and continue to Solicit for the unfulfilled IA options, or
3. The client could continue with the single session, and include the unfulfilled IA options in subsequent messages to the server.

Resetting the state machine and continuing to send Solicit messages may result in the client never completing DHCP and is generally not considered a good solution. It can also result in a packet storm if the client does not appropriately rate limit its sending of Solicit messages or there are many clients on the network. Client implementors that follow this approach, SHOULD implement the updates to RFC-3315 specified in [RFC7083].

Creating a separate DHCP session (separate instances of the client state machine) per IA option type, while conceptually simple, causes a number of issues: additional host resources required to create and maintain multiple instances of the state machine in clients, additional DHCP protocol traffic, unnecessary duplication of other configuration options and the potential for conflict, divergence in

that each IA option type specification specifies its 'own' version of the DHCP protocol.

The single session and state machine allows the client to use the best configuration it is able to obtain from a single DHCP server during the configuration exchange. Note, however, that the server may not be configured to deliver the entire configuration requested by the client. In that case the client could continue to operate only using the configuration received, even if other servers can provide the missing configuration. In practice, especially in the case of handling IA_NA and IA_PD, this situation should be rare or a temporary operational error. So, it is more likely for the client to get all configuration if it continues, in each subsequent configuration exchange, to request all the configuration information it is programmed to try to obtain, including any stateful configuration options for which no results were returned in previous exchanges.

One major issue of this last approach is that it is difficult to allow it with the current DHCPv6 specifications; in some cases they are not clear enough, and in other cases existing restrictions can make it impossible. This document introduces some clarifications and small modifications to the current specifications to address these concerns.

While all approaches have their own pros and cons, approach 3 SHOULD be used and is the focus of this document because it is deemed to work best for common cases of the mixed use of IA_NA and IA_PD. But this document does not exclude other approaches. Also, in some corner cases it may not be feasible to maintain a single DHCPv6 session for both IA_NA and IA_PD. These corner cases are beyond the scope of this document and may depend on the network in which the client (CER) is designed to operate and on the functions the client is required to perform.

The sections which follow update RFC 3315 and RFC 3633 to accommodate the recommendation, though many of the changes are also applicable even if other approaches are used.

4.1. Placement of Status Codes in an Advertise Message

In Reply messages IA specific status codes (i.e., NoAddrsAvail, NotOnLink, NoBinding, NoPrefixAvail) are encapsulated in the IA option. In Advertise messages though, the NoAddrsAvail code is returned at in the top level. This makes sense if the client is only interested in the assignment of the addresses and the failure case is fatal. However, if the client sends both IA_NA and IA_PD options in a Solicit message, it is possible that the server offers no addresses

but it offers some prefixes, and the client may choose to send a Request message to obtain the offered prefixes. In this case, it is better if the Status Code option for IA specific status codes is encapsulated in the IA option to indicate that the failure occurred for the specific IA. This also makes the NoAddrsAvail and NoPrefixAvail Status Code option placement for Advertise messages identical to Reply messages.

In addition, how a server formats the Advertise message when addresses are not available has been a point of some confusion and implementations seem to vary (some strictly follow RFC 3315 while others assumed it was encapsulated in the IA option as for Reply messages).

We have chosen the following solution:

Clients MUST handle each of the following Advertise messages formats when there are no addresses available (even when no other IA option types were in the Solicit):

1. Advertise containing the IA_NAs and/or IA_TAs with encapsulated Status Code option of NoAddrsAvail and no top-level Status Code option.
2. Advertise containing just a top-level Status Code option of NoAddrsAvail and no IA_NAs/IA_TAs.
3. Advertise containing a top-level Status Code option of NoAddrsAvail and IA_NAs and/or IA_TAs with a Status Code option of NoAddrsAvail.

Note: Clients MUST handle the last two formats listed above to facilitate backward compatibility with the servers which have not been updated to this specification.

See Section 4.2 for updated text for Section 17.1.3 of RFC 3315 and Section 11.1 of RFC 3633.

Servers MUST return the Status Code option of NoAddrsAvail encapsulated in IA_NA/IA_TA options and MUST NOT return a top-level Status Code option of NoAddrsAvail when no addresses will be assigned (1 in the above list). This means that the Advertise response matches the Reply response with respect to the handling of the NoAddrsAvail status.

Replace the following paragraph in RFC 3315, section 17.2.2:

If the server will not assign any addresses to any IAs in a subsequent Request from the client, the server MUST send an Advertise message to the client that includes only a Status Code option with code NoAddrsAvail and a status message for the user, a Server Identifier option with the server's DUID, and a Client Identifier option with the client's DUID.

With:

If the server will not assign any addresses to an IA in a subsequent Request from the client, the server MUST include the IA in the Advertise message with no addresses in the IA and a Status Code option encapsulated in the IA containing status code NoAddrsAvail.

4.2. Advertise Message Processing by a Client

[RFC3315] specifies that a client must ignore an Advertise message if a server will not assign any addresses to a client, and [RFC3633] specifies that a client must ignore an Advertise message if a server returns the NoPrefixAvail status to a requesting router. Thus, a client requesting both IA_NA and IA_PD, with a server that only offers either addresses or delegated prefixes, is not supported by the current protocol specifications.

Solution: a client SHOULD accept Advertise messages, even when not all IA option types are being offered. And, in this case, the client SHOULD include the not offered IA option types in its Request. A client SHOULD only ignore an Advertise message when none of the requested IA options include offered addresses or delegated prefixes. Note that ignored messages MUST still be processed for SOL_MAX_RT and INF_MAX_RT options as specified in [RFC7083].

Replace Section 17.1.3 of RFC 3315: (existing errata)

The client MUST ignore any Advertise message that includes a Status Code option containing the value NoAddrsAvail, with the exception that the client MAY display the associated status message(s) to the user.

With (this includes the changes made by [RFC7083]):

The client MUST ignore any Advertise message that contains no addresses (IAADDR options encapsulated in IA_NA or IA_TA options) and no delegated prefixes (IAPREFIX options encapsulated in IA_PD options, see RFC 3633) with the exception that the client:

- MUST process an included SOL_MAX_RT option (RFC 7083) and
- MUST process an included INF_MAX_RT option (RFC 7083).

A client can display any associated status message(s) to the user or activity log.

The client ignoring this Advertise message MUST NOT restart the Solicit retransmission timer.

And, replace:

- The client MAY choose a less-preferred server if that server has a better set of advertised parameters, such as the available addresses advertised in IAs.

With:

- The client MAY choose a less-preferred server if that server has a better set of advertised parameters, such as the available set of IAs, as well as the set of other configuration options advertised.

And, replace the last paragraph of Section 11.1 of RFC 3633 with:

The requesting router MUST ignore any Advertise message that contains no addresses (IAADDR options encapsulated in IA_NA or IA_TA options) and no delegated prefixes (IAPREFIX options encapsulated in IA_PD options, see RFC 3633) with the exception that the requesting router:

- MUST process an included SOL_MAX_RT option (RFC 7083) and
- MUST process an included INF_MAX_RT option (RFC 7083).

A client can display any associated status message(s) to the user or activity log.

The requesting router ignoring this Advertise message MUST NOT restart the Solicit retransmission timer.

4.3. T1/T2 Timers

The T1 and T2 times determine when the client will contact the server to extend lifetimes of information received in an IA. How should a client handle the case where multiple IA options have different T1 and T2 times?

In a multiple IA option type model, the T1/T2 times are protocol timers, that should be independent of the IA options themselves. If we were to redo the DHCP protocol from scratch the T1/T2 times should be carried in a separate DHCP option.

Solution: The server MUST set the T1/T2 times in all IA options in a Reply or Advertise message to the same value. To deal with the case where servers have not yet been updated to do that, the client MUST select a T1 and T2 time from all IA options which will guarantee that the client will send Renew/Rebind messages not later than at the T1/T2 times associated with any of the client's bindings.

As an example, if the client receives a Reply with T1_NA of 3600 / T2_NA of 5760 and T1_PD of 0 / T2_PD of 1800, the client SHOULD use the T1_PD of 0 / T2_PD of 1800. The reason for this is that a T1 of 0 means that the Renew time is at the client's discretion, but this value cannot be greater than the T2 value (1800).

The following paragraph should be added to Sections 18.2.1, 18.2.3, and 18.2.4 of RFC 3315:

The T1/T2 times set in each applicable IA option for a Reply MUST be the same values across all IAs. The server MUST determine the T1/T2 times across all of the applicable client's bindings in the Reply. This facilitates the client being able to renew all of the bindings at the same time.

Note: This additional paragraph has also been included in the revised text later for Sections 18.2.3 and 18.2.4 of RFC 3315.

Changes for client T1/T2 handling are included in Section 4.4.3 and Section 4.4.4.

4.4. Renew and Rebind Messages

This section presents issues with handling multiple IA option types in the context of creation and processing the Renew and Rebind messages. It also introduces relevant updates to the [RFC3315] and [RFC3633].

4.4.1. Renew Message

In multiple IA option type model, the client may include multiple IA options in the Request message, and the server may create bindings only for a subset of the IA options included by the client. For the IA options in the Request message for which the server does not create the bindings, the server sends the IA options in the Reply message with the NoAddrsAvail or NoPrefixAvail status codes.

The client may accept the bindings created by the server, but may desire the other bindings to be created once they become available, e.g. when the server configuration is changed. The client which accepted the bindings created by the server will periodically send a Renew message to extend their lifetimes. However, the Renew message, as described in the [RFC3315], does not support the ability for the client to extend the lifetimes of the bindings for some IAs, while requesting bindings for other IAs.

Solution: The client, which sends a Renew message to extend the lifetimes of the bindings assigned to the client, SHOULD include IA options for these bindings as well as IA options for all other bindings that the client desires but has been unable to obtain. The client and server processing need to be modified. Note that this change makes the server's IA processing of Renew similar to the Request processing.

4.4.2. Rebind Message

According to the Section 4.4.1, the client includes IA options in a Renew message for the bindings it desires but has been unable to obtain by sending a Request message, apart from the IA options for the existing bindings.

At time T2, the client stops sending Renew messages to the server and initiates the Rebind/Reply message exchange with any available server. In this case, it should be possible to continue trying to obtain new bindings using the Rebind message if the client failed to get the response from the server to the Renew message.

Solution: The client SHOULD continue to include the IA options received from the server and it MAY include additional IA options to request creation of the additional bindings.

4.4.3. Updates to section 18.1.3 of RFC 3315

Replace Section 18.1.3 of RFC 3315 with the following text:

To extend the valid and preferred lifetimes for the addresses assigned to an IA, the client sends a Renew message to the server from which the addresses were obtained, which includes an IA option for the IA whose address lifetimes are to be extended. The client includes IA Address options within the IA option for the addresses assigned to the IA. The server determines new lifetimes for these addresses according to the administrative configuration of the server. The server may also add new addresses to the IA. The server can remove addresses from the IA by returning IA Address

options for such addresses with preferred and valid lifetimes set to zero.

The server controls the time at which the client contacts the server to extend the lifetimes on assigned addresses through the T1 and T2 parameters assigned to an IA. However, as the client Renews/Rebinds all IAs from the server at the same time, the client MUST select a T1 and T2 time from all IA options which will guarantee that the client will send Renew/Rebind messages not later than at the T1/T2 times associated with any of the client's bindings.

At time T1, the client initiates a Renew/Reply message exchange to extend the lifetimes on any addresses in the IA.

If T1 or T2 had been set to 0 by the server (for an IA_NA) or there are no T1 or T2 times (for an IA_TA) in a previous Reply, the client may send a Renew or Rebind message, respectively, at the client's discretion.

The client sets the "msg-type" field to RENEW. The client generates a transaction ID and inserts this value in the "transaction-id" field.

The client places the identifier of the destination server in a Server Identifier option.

The client MUST include a Client Identifier option to identify itself to the server. The client adds any appropriate options, including one or more IA options.

For IAs to which addresses have been assigned, the client includes a corresponding IA option containing an IA Address option for each address assigned to the IA. The client MUST NOT include addresses in any IA option that the client did not obtain from the server or that are no longer valid (that have a zero valid lifetime).

The client MAY include an IA option for each binding it desires but has been unable to obtain. This IA option MUST NOT contain any addresses. However, it MAY contain the IA Address option with IPv6 address field set to 0 to indicate the client's preference for the preferred and valid lifetimes for any newly assigned addresses.

The client MUST include an Option Request option (see section 22.7) to indicate the options the client is interested in receiving. The client MAY include options with data values as hints to the server about parameter values the client would like to have returned.

The client transmits the message according to section 14, using the following parameters:

IRT	REN_TIMEOUT
MRT	REN_MAX_RT
MRC	0
MRD	Remaining time until T2

The message exchange is terminated when time T2 is reached (see section 18.1.4), at which time the client begins a Rebind message exchange.

4.4.4. Updates to Section 18.1.4 of RFC 3315

Replace Section 18.1.4 of RFC 3315 with the following text:

At time T2 (which will only be reached if the server to which the Renew message was sent at time T1 has not responded), the client initiates a Rebind/Reply message exchange with any available server.

The client constructs the Rebind message as described in 18.1.3 with the following differences:

- The client sets the "msg-type" field to REBIND.
- The client does not include the Server Identifier option in the Rebind message.

The client transmits the message according to section 14, using the following parameters:

IRT	REB_TIMEOUT
MRT	REB_MAX_RT
MRC	0
MRD	Remaining time until valid lifetimes of all addresses in all IAs have expired

If all addresses for an IA have expired the client may choose to include this IA without any addresses (or with only a hint for lifetimes) in subsequent Rebind messages to indicate that the client is interested in assignment of the addresses to this IA.

The message exchange is terminated when the valid lifetimes of all addresses across all IAs have expired, at which time the client uses Solicit message to locate a new DHCP server and sends a Request for the expired IAs to the new server.

4.4.5. Updates to Section 18.1.8 of RFC 3315

Replace Section 18.1.8 of RFC 3315 with the following text:

Upon the receipt of a valid Reply message in response to a Solicit (with a Rapid Commit option), Request, Confirm, Renew, Rebind or Information-request message, the client extracts the configuration information contained in the Reply. The client MAY choose to report any status code or message from the status code option in the Reply message.

If the client receives a Reply message with a Status Code containing UnspecFail, the server is indicating that it was unable to process the message due to an unspecified failure condition. If the client retransmits the original message to the same server to retry the desired operation, the client MUST limit the rate at which it retransmits the message and limit the duration of the time during which it retransmits the message.

When the client receives a Reply message with a Status Code option with the value UseMulticast, the client records the receipt of the message and sends subsequent messages to the server through the interface on which the message was received using multicast. The client resends the original message using multicast.

When the client receives a NotOnLink status from the server in response to a Confirm message, the client performs DHCP server solicitation, as described in section 17, and client-initiated configuration as described in section 18. If the client receives any Reply messages that do not indicate a NotOnLink status, the client can use the addresses in the IA and ignore any messages that indicate a NotOnLink status.

When the client receives a NotOnLink status from the server in response to a Request, the client can either re-issue the Request without specifying any addresses or restart the DHCP server discovery process (see section 17).

The client SHOULD perform duplicate address detection [17] on each of the received addresses in any IAs, on which it has not performed duplicate address detection during processing of any of the previous Reply messages from the server. The client performs the duplicate address detection before using the received addresses for

the traffic. If any of the addresses are found to be in use on the link, the client sends a Decline message to the server for those addresses as described in section 18.1.7.

If the Reply was received in response to a Solicit (with a Rapid Commit option), Request, Renew or Rebind message, the client updates the information it has recorded about IAs from the IA options contained in the Reply message:

- Record T1 and T2 times.
- Add any new addresses in the IA option to the IA as recorded by the client.
- Update lifetimes for any addresses in the IA option that the client already has recorded in the IA.
- Discard any addresses from the IA, as recorded by the client, that have a valid lifetime of 0 in the IA Address option.
- Leave unchanged any information about addresses the client has recorded in the IA but that were not included in the IA from the server.

Management of the specific configuration information is detailed in the definition of each option in section 22.

The client examines the status code in each IA individually. If the client receives a NoAddrsAvail status code, the client has received no usable addresses in the IA.

If the client can operate with the addresses obtained from the server the client uses addresses and other information from any IAs that do not contain a Status Code option with the NoAddrsAvail status code. The client MAY include the IAs for which it received the NoAddrsAvail status code, with no addresses, in subsequent Renew and Rebind messages sent to the server, to retry obtaining the addresses for these IAs.

If the client cannot operate without the addresses for the IAs for which it received the NoAddrsAvail status code, the client may try another server (perhaps by restarting the DHCP server discovery process).

If the client finds no usable addresses in any of the IAs, it may either try another server (perhaps restarting the DHCP server discovery process) or use the Information-request message to obtain other configuration information only.

When the client receives a Reply message in response to a Renew or Rebind message, the client:

- sends a Request message if any of the IAs in the Reply message contains the NoBinding status code. The client places IA options in this message for only those IAs for which the server returned the NoBinding status code in the Reply message. The client continues to use other bindings for which the server did not return an error
- sends a Renew/Rebind if any of the IAs is not in the Reply message, but in this case the client MUST limit the rate at which it sends these messages, to avoid the Renew/Rebind storm
- otherwise accepts the information in the IA.

When the client receives a valid Reply message in response to a Release message, the client considers the Release event completed, regardless of the Status Code option(s) returned by the server.

When the client receives a valid Reply message in response to a Decline message, the client considers the Decline event completed, regardless of the Status Code option(s) returned by the server.

4.4.6. Updates to Section 18.2.3 of RFC 3315

Replace Section 18.2.3 of RFC 3315 with the following text:

When the server receives a Renew message via unicast from a client to which the server has not sent a unicast option, the server discards the Renew message and responds with a Reply message containing a Status Code option with the value UseMulticast, a Server Identifier option containing the server's DUID, the Client Identifier option from the client message, and no other options.

For each IA in the Renew message from a client, the server locates the client's binding and verifies that the information in the IA from the client matches the information stored for that client.

If the server finds the client entry for the IA the server sends back the IA to the client with new lifetimes and, if applicable, T1/T2 times. If the server is unable to extend the lifetimes of an address in the IA, the server MAY choose not to include the IA Address option for this address.

The server may choose to change the list of addresses and the lifetimes of addresses in IAs that are returned to the client.

If the server finds that any of the addresses in the IA are not appropriate for the link to which the client is attached, the server returns the address to the client with lifetimes of 0.

For each IA for which the server cannot find a client entry, the server has the following choices depending on the server's policy and configuration information:

- If the server is configured to create new bindings as a result of processing Renew messages, the server SHOULD create a binding and return the IA with allocated addresses with lifetimes and, if applicable, T1/T2 times and other information requested by the client. The server MAY use values in the IA Address option (if included) as a hint.
- If the server is configured to create new bindings as a result of processing Renew messages, but the server will not assign any addresses to an IA, the server returns the IA option containing a Status Code option with the NoAddrsAvail status code and a status message for a user.
- If the server does not support creation of new bindings for the client sending a Renew message, or if this behavior is disabled according to the server's policy or configuration information, the server returns the IA option containing a Status code option with the NoBinding status code and a status message for a user.

The server constructs a Reply message by setting the "msg-type" field to REPLY, and copying the transaction ID from the Renew message into the transaction-id field.

The server MUST include a Server Identifier option containing the server's DUID and the Client Identifier option from the Renew message in the Reply message.

The server includes other options containing configuration information to be returned to the client as described in section 18.2.

The T1/T2 times set in each applicable IA option for a Reply MUST be the same values across all IAs. The server MUST determine the T1/T2 times across all of the applicable client's bindings in the Reply. This facilitates the client being able to renew all of the bindings at the same time.

4.4.7. Updates to Section 18.2.4 of RFC 3315

Replace Section 18.2.4 of RFC 3315 with the following text:

When the server receives a Rebind message that contains an IA option from a client, it locates the client's binding and verifies that the information in the IA from the client matches the information stored for that client.

If the server finds the client entry for the IA and the server determines that the addresses in the IA are appropriate for the link to which the client's interface is attached according to the server's explicit configuration information, the server SHOULD send back the IA to the client with new lifetimes and, if applicable, T1/T2 times. If the server is unable to extend the lifetimes of an address in the IA, the server MAY choose not to include the IA Address option for this address.

If the server finds the client entry for the IA and any of the addresses are no longer appropriate for the link to which the client's interface is attached according to the server's explicit configuration information, the server returns the address to the client with lifetimes of 0.

If the server cannot find a client entry for the IA, the IA contains addresses and the server determines that the addresses in the IA are not appropriate for the link to which the client's interface is attached according to the server's explicit configuration information, the server MAY send a Reply message to the client containing the client's IA, with the lifetimes for the addresses in the IA set to 0. This Reply constitutes an explicit notification to the client that the addresses in the IA are no longer valid. In this situation, if the server does not send a Reply message it silently discards the Rebind message.

Otherwise, for each IA for which the server cannot find a client entry, the server has the following choices depending on the server's policy and configuration information:

- If the server is configured to create new bindings as a result of processing Rebind messages (also see the note about the Rapid Commit option below), the server SHOULD create a binding and return the IA with allocated addresses with lifetimes and, if applicable, T1/T2 times and other information requested by the client. The server MAY use values in the IA Address option (if included) as a hint.

- If the server is configured to create new bindings as a result of processing Rebind messages, but the server will not assign any addresses to an IA, the server returns the IA option containing a Status Code option with the NoAddrsAvail status code and a status message for a user.
- If the server does not support creation of new bindings for the client sending a Rebind message, or if this behavior is disabled according to the server's policy or configuration information, the server returns the IA option containing a Status Code option with the NoBinding status code and a status message for a user.

When the server creates new bindings for the IA it is possible that other servers also create bindings as a result of receiving the same Rebind message. This is the same issue as in the Discussion under the Rapid Commit option, see section 22.14. Therefore, the server SHOULD only create new bindings during processing of a Rebind message if the server is configured to respond with a Reply message to a Solicit message containing the Rapid Commit option.

The server constructs a Reply message by setting the "msg-type" field to REPLY, and copying the transaction ID from the Rebind message into the transaction-id field.

The server MUST include a Server Identifier option containing the server's DUID and the Client Identifier option from the Rebind message in the Reply message.

The server includes other options containing configuration information to be returned to the client as described in section 18.2.

The T1/T2 times set in each applicable IA option for a Reply MUST be the same values across all IAs. The server MUST determine the T1/T2 times across all of the applicable client's bindings in the Reply. This facilitates the client being able to renew all of the bindings at the same time.

4.4.8. Updates to RFC 3633

Replace the following text in Section 12.1 of RFC 3633:

Each prefix has valid and preferred lifetimes whose durations are specified in the IA_PD Prefix option for that prefix. The requesting router uses Renew and Rebind messages to request the extension of the lifetimes of a delegated prefix.

With:

Each prefix has valid and preferred lifetimes whose durations are specified in the IA_PD Prefix option for that prefix. The requesting router uses Renew and Rebind messages to request the extension of the lifetimes of a delegated prefix.

The requesting router MAY include IA_PD options without any prefixes, i.e. without IA Prefix option or with IPv6 prefix field of IA Prefix option set to 0, in a Renew or Rebind message to obtain bindings it desires but has been unable to obtain. The requesting router MAY set the prefix-length field of the IA Prefix option as a hint to the server. As in [RFC3315], the requesting router MAY also provide lifetime hints in the IA Prefix option.

Replace the following text in Section 12.2 of RFC 3633:

The delegating router behaves as follows when it cannot find a binding for the requesting router's IA_PD:

With:

For the Renew or Rebind, if the IA_PD contains no IA Prefix option or it contains an IA Prefix option with the IPv6 prefix field set to 0, the delegating router SHOULD assign prefixes to the IA_PD according to the delegating router's explicit configuration information. In this case, if the IA_PD contains an IA Prefix option with the IPv6 prefix field set to 0, the delegating router MAY use the value in the prefix-length field of the IA Prefix option as a hint for the length of the prefixes to be assigned. The delegating router MAY also respect lifetime hints provided by the requesting router in the IA Prefix option.

The delegating router behaves as follows when it cannot find a binding for the requesting router's IA_PD containing prefixes:

4.5. Confirm Message

The Confirm message, as described in [RFC3315], is specific to address assignment. It allows a server without a binding to reply to the message, under the assumption that the server only needs knowledge about the prefix(es) on the link, to inform the client that the address is likely valid or not. This message is sent when e.g. the client has moved and needs to validate its addresses. Not all bindings can be validated by servers and the Confirm message provides for this by specifying that a server that is unable to determine the on-link status MUST NOT send a Reply.

Note: Confirm has a specific meaning and does not overload Renew/Rebind. It also is lower processing cost as the server does NOT need to extend lease times or otherwise send back other configuration options.

The Confirm message is used by the client to verify that it has not moved to a different link. For IAs with addresses, the mechanism used to verify if a client has moved or not, is by matching the link's on-link prefix(es) (typically a /64) against the prefix-length first bits of the addresses provided by the client in the IA_NA or IA_TA IA-types. As a consequence Confirm can only be used when the client has an IA with address(es) (IA_NA or IA_TA).

A client MUST have a binding including an IA with addresses to use the Confirm message. A client with IAs with addresses as well as other IA-types MAY, depending on the IA-type, use the Confirm message to detect if the client has moved to a different link. A client that does not have a binding with an IA with addresses MUST use the Rebind message instead.

IA_PD requires verification that the delegating router (server) has the binding for the IAs. In that case a requesting router (client) MUST use the Rebind message in place of the Confirm message and it MUST include all of its bindings, even address IAs.

Note that Section 18.1.2 of RFC 3315 states that a client MUST initiate a Confirm when it may have moved to a new link. This is relaxed to a SHOULD as a client may have determined whether it has or has not moved using other techniques, such as described in [RFC6059]. And, as stated above, a client with delegated prefixes, MUST send a Rebind instead of a Confirm.

4.6. Decline Should Not Necessarily Trigger a Release

Some client implementations have been found to send a Release message for other bindings they may have received after they determine a conflict and have correctly sent a Decline message for the conflicting address(es).

A client SHOULD NOT send a Release message for other bindings it may have received just because it sent a Decline message. The client SHOULD retain the non-conflicting bindings. The client SHOULD treat the failure to acquire a binding as a result of the conflict, to be equivalent to not having received the binding, insofar as it behaves when sending Renew and Rebind messages.

4.7. Multiple Provisioning Domains

This document has assumed that all DHCP servers on a network are in a single provisioning domain and thus should be "equal" in the service that they offer. This was also assumed by [RFC3315] and [RFC3633].

One could envision a network where the DHCP servers are in multiple provisioning domains, and it may be desirable to have the DHCP client obtain different IA types from different provisioning domains. How a client detects the multiple provisioning domains and how it would interact with the multiple servers in these different domains is outside the scope of this document (see [I-D.ietf-mif-mpvd-arch] and [I-D.ietf-mif-mpvd-dhcp-support]).

5. IANA Considerations

This specification does not require any IANA actions.

6. Security Considerations

There are no new security considerations pertaining to this document.

7. Acknowledgements

Thanks to many people that contributed to identify the stateful issues addressed by this document and for reviewing drafts of the document, including Ralph Droms, John Brzozowski, Ted Lemon, Hemant Singh, Wes Beebe, Gaurau Halwasia, Bud Millword, Tim Winters, Rob Shakir, Jinmei Tatuya, Andrew Yourtchenko, Fred Templin, Tomek Mrugalski, Suresh Krishnan, and Ian Farrer.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC7083] Droms, R., "Modification to Default Values of SOL_MAX_RT and INF_MAX_RT", RFC 7083, November 2013.

8.2. Informative References

- [I-D.dhcgwg-dhc-rfc3315bis]
Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A.,
Richardson, M., Jiang, S., and T. Lemon, "Dynamic Host
Configuration Protocol for IPv6 (DHCPv6) bis", draft-
dhcgwg-dhc-rfc3315bis-04 (work in progress), February 2015.
- [I-D.ietf-mif-mpvd-arch]
Anipko, D., "Multiple Provisioning Domain Architecture",
draft-ietf-mif-mpvd-arch-11 (work in progress), March
2015.
- [I-D.ietf-mif-mpvd-dhcp-support]
Krishnan, S., Korhonen, J., and S. Bhandari, "Support for
multiple provisioning domains in DHCPv6", draft-ietf-mif-
mpvd-dhcp-support-01 (work in progress), March 2015.
- [RFC6059] Krishnan, S. and G. Daley, "Simple Procedures for
Detecting Network Attachment in IPv6", RFC 6059, November
2010.
- [RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic
Requirements for IPv6 Customer Edge Routers", RFC 7084,
November 2013.
- [RFC7227] Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and
S. Krishnan, "Guidelines for Creating New DHCPv6 Options",
BCP 187, RFC 7227, May 2014.

Authors' Addresses

Ole Troan
Cisco Systems, Inc.
Philip Pedersens vei 20
N-1324 Lysaker
Norway

Email: ot@cisco.com

Bernie Volz
Cisco Systems, Inc.
1414 Massachusetts Ave
Boxborough, MA 01719
USA

Email: volz@cisco.com

Marcin Siodelski
ISC
950 Charter Street
Redwood City, CA 94063
USA

Email: msiodelski@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2013

S. Jiang, Ed.
F. Xia
B. Sarikaya
Huawei Technologies
February 25, 2013

Prefix Assignment in DHCPv6
draft-ietf-dhc-host-gen-id-05

Abstract

This document introduces a generic host-oriented prefix assignment mechanism using DHCPv6. In this new address configuration procedure, the prefix is assigned from a DHCPv6 server to hosts through DHCPv6 message exchanging while the interface identifiers are independently generated by the hosts. It enables both integral address assignment and self-generated addresses in one single mechanism, DHCPv6. It also enables stateless address configuration without RA attendance. The technique described in this document can be used in networks which assign IPv6 addresses using DHCPv6, e.g. WiMAX.

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, 2013.

Copyright Notice

Copyright (c) 2013 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. Applicability	4
4. Address Auto-configuration	5
5. DHCPv6 Operation	5
6. DHCPv6 IA_PA Option	7
6.1. Identity Association for Prefix Assignment Option	7
6.2. IA_PA Prefix Option	9
7. IANA consideration	9
8. Security Considerations	9
9. Acknowledgements	9
10. References	9
10.1. Normative References	9
10.2. Informative references	10
Authors' Addresses	11

1. Introduction

A host IPv6 address is combined by a prefix and an interface identifier. Currently, there are two mechanisms to configure a host IPv6 address. [RFC3315] describes the operation of address assignment by a DHCPv6 server. The operation assumes that the server is responsible for the assignment of an integral address which includes both prefix and interface identifier parts as described in [RFC4291]. In the Stateless Address Autoconfiguration (SLAAC, [RFC4862]) model, the interface Identifier is generated by the host itself while the prefix is configured through Router Advertisement message defined in [RFC4861].

However, in a DHCPv6-managed network, assigning 128-bit address is insufficient. Some hosts may want to use self-generated address, which are combined by prefixes obtained from network configuration and interface identifiers generated by hosts. The applicable user cases include CGA [RFC3972], modified EUI-64 interface identifier [EUI-64], temporary addresses for privacy [RFC4941] and etc.

In these scenarios, the address configuration procedure has to be splitted in two methods: integral address assignment through DHCPv6 and prefix announcement by RA advertisement. Some ISPs desire to manage address configuration using one set of protocol, rather than mixture of DHCPv6 and Neighbor Discovery.

There are also some network environments in that prefix announcement through RAs may not be the best choice. For example, hosts may connect through tunnels, either layer 2 tunnels or layer 3 tunnels.

While a RA is only able to announce prefix on a single link, DHCPv6 configuration can be used to manage multiple links by setup DHCPv6 relay.

Up to now, there is no mechanism for host-oriented prefix assignment in DHCPv6. [RFC3633] defines Prefix Delegation options providing a mechanism for automated delegation of IPv6 prefixes using the DHCPv6. This mechanism is intended for delegating a long-lived prefix from a delegating router to a requesting router. This mechanism "is not bound to the assignment of IP addresses or other configuration information to hosts" [RFC3633]. It delegates prefixes to a routable device for itself use only. It does not support the host-generated interface identifiers model, in which prefix(es) need to be propagated to hosts.

This document introduces a generic prefix assignment mechanism using DHCPv6. In this new address configuration procedure, the prefix is propagated from a DHCPv6 server to hosts through DHCPv6 message

exchanging while the interface identifiers are independently generated by the hosts. It enables both integral address assignment and self-generated addresses in one single mechanism, DHCPv6. Note, in many scenarios, Neighbor Discovery [RFC4861] is still needed for routing and reachability. In other scenarios, this mechanism enables stateless address configuration while RA absents.

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].

The terminology in this document is mainly based on the definitions in [RFC3315] and [RFC3633].

Prefix assignment: a DHCPv6 server propagates prefix information to hosts in unicast model.

3. Applicability

In point-to-point link model, DHCPv6 operation with host-generated interface identifier, described in this document, may be used. [RFC4968] provides different IPv6 link models that are suitable for 802.16 based networks and a point-to-point link model is recommended. Also, 3GPP and 3GPP2 have earlier adopted the point-to-point link model based on the recommendations in [RFC3314]. In this model, one prefix can only be assigned to one interface of a host (mobile station) and different hosts (mobile stations) can't share a prefix. The unique prefix can be used to identify the host. It is not necessary for a DHCPv6 server to generate an interface identifier for the host. The host may generate its interface identifier as described in [RFC4941]. An interface identifier could even be generated via random number generation.

[RFC3972] defines Cryptographically Generated Addresses (CGA), which is generated from a giving prefix and a public signature key. For security reasons, it is only proper to be generated the user, the host itself. It requests a prefix before the interface identifier can be computed.

Modified EUI-64 interface identifier [EUI-64] is also typically generated by hosts. [RFC4941] has defined temporary addresses for privacy purposes. The temporary addresses is also generated by hosts using random algorithm.

The DHCPv6 operations defined in this document supports abovementioned address methods, and the host-generated addresses that may defined in the future.

4. Address Auto-configuration

Router Advertisements in ND [RFC4861] allow routers to inform hosts how to perform Address Auto-configuration. For example, routers can specify whether hosts should use DHCPv6 and/or stateless address configuration. In Router Advertisement message, M and O bits are used for indication of address auto-configuration mode.

Whatever address auto-configuration mode a host uses, the following two parts are necessary for the host to formulate it's IPv6 address:

- o A prefix. "A bit string that consists of some number of initial bits of an address" [RFC4861]. The prefixes can be announced through Router Advertisement message. Prefix assignment from a DHCPv6 server is not currently support.
- o An interface identifier. "From address autoconfiguration's perspective, an interface identifier is a bit string of known length" [RFC4862]. Modified EUI-64 interface identifier [EUI-64] is a widely-used host generated interface identifier. It generates interface identifier from the host MAC address. The interface identifier of CGA [RFC3972] is generated by computing a preifx that will be used to form the CGA and a cryptographic hash of a public key of a host. The host is responsible for interface identifier generation.

In the ND-managed environment, RA is used to assign the prefix.

So far, there is no mechanism to support the scenario that prefixes are managed by a DHCPv6 server. This document targets to meet this gap. The DHCPv6 operation defined in this document enables the DHCPv6 server to assign a prefix, rather than a integral address, to the host, so that the host can obtain an IPv6 address by combining the prefix with its own generated interface identifier. It enables the auto address configuration through DHCPv6.

5. DHCPv6 Operation

Figure 1 shows the operation of separating prefix assignment and interface identifier generation in the DHCPv6.

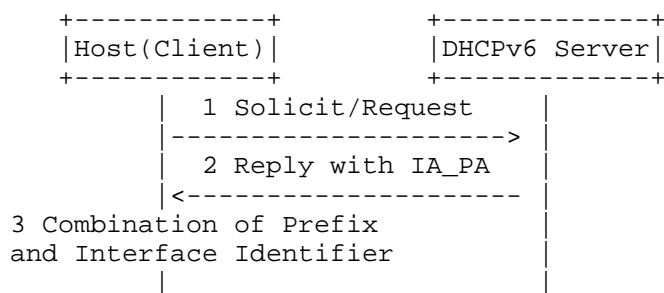


Figure 1: DHCPv6 Operation

1. A host uses a Solicit message to discover DHCPv6 servers. Indications of information requests can be included in the Solicit message or a Request message after discovery procedure. If a host that wants to use host generated addresses, it SHOULD request prefix assignment explicitly by including an IA_PA in a Solicit or a Request message, in which an IAID is provided by the host.
2. The DHCPv6 server assigns one or more prefixes to the host in the Reply messages responding to the prefix requests from the hosts. A server MUST return the same set of prefixes for the same IA_PA (as identified by the IAID) as long as those prefixes are still valid. After the lifetimes of the prefixes in an IA_TA have expired, the IAID may be reused to identify a new IA_PA with new prefix. If there is not a proper prefix available, a NoPrefixAvail (defined in [RFC3633]) status-code is returned to the host and the procedure is terminated.
3. The host generates an interface identifier and formulates a combined IPv6 address by concatenating the assigned prefix and the self-generated interface identifier.

After the host generates an IPv6 address using the above procedure, the host may send a Request message to the DHCPv6 server in order to confirm the usage of the new address. The confirmation procedure may be completed together with the address registration procedure [I-D.ietf-dhc-addr-registration]. However, the confirmation procedure is out of scope.

When the host reaches T1 or T2 defined in Section 6.1, it SHOULD use the same message exchanges, as described in section 18, "DHCP Client-Initiated Configuration Exchange" of [RFC3315], to obtain or update prefix(es) from a DHCPv6 server.

A DHCPv6 server MAY initiatively send a reconfiguration message to the host, as described in section 19, "DHCP Server-Initiated Configuration Exchange" of [RFC3315], to cause prefix(es) information

update.

If an IA_PA capable client connects to a network, and the DHCPv6 server is not IA_PA capable, the Solicit or Request message with IA_PA Option will result in no Reply, Reply without IA_PAs, or Reply with a Status Code containing UnspecFail. The client MAY decide the network does not support IA_PA immediately or after a period of soliciting (with limited retransmissions times). Then, it MAY "failover" to IA_NA/IA_TA requests.

6. DHCPv6 IA_PA Option

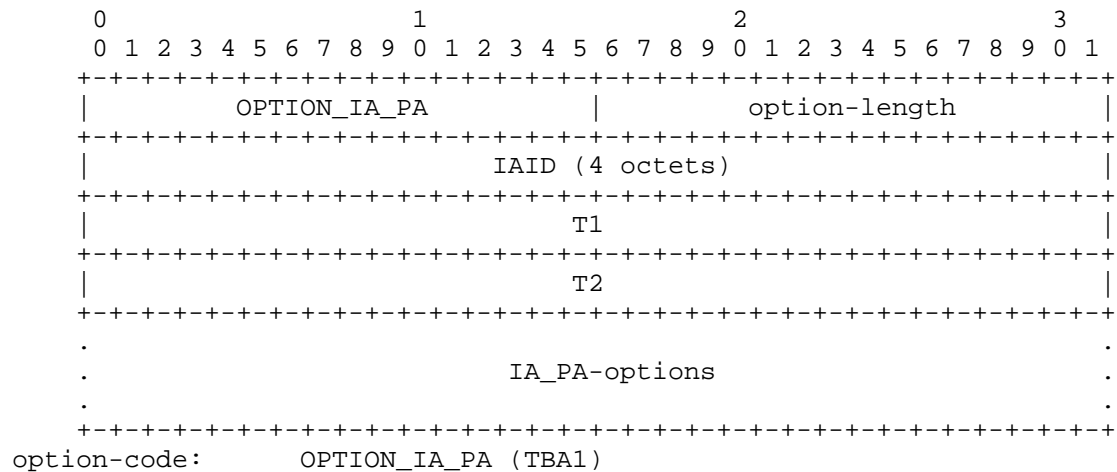
In this section, one new option is defined, Identity Association for Prefix Assignment Option . The format of this new DHCPv6 IA_PA Option has been deliberately designed to be the same with IA_PD option[RFC3633]. The IA_PD Prefix and IA Address sub-options from IA_PD option are also reused. However, the two options are different on the semantics and usage models.

Comparing with Prefix Information Option in ND, Section 4.6.2 of [RFC4861], the IA_PA option does not provide L flag and A flag. The A (autonomous address-configuration flag) isn't need obviously because the IA_PA is implicit for stateless address configuration. Because the IA_PA is only address relevant, it does not relevant to reachability or routing and the DHCPv6 server may not sure the on-link state. So L (on-Link) flag is not include. The DHCPv6 client should treat the prefix as same as L flag not set, which makes no statement about on-link or off-link properties of the prefix.

6.1. Identity Association for Prefix Assignment Option

The IA_PA option is used to carry a prefix assignment identity association, the parameters associated with the IA_PA and the prefixes associated with it.

The format of the IA_PA option is:



option-length: 12 + length of IA_PA-options field.

IAID: The unique identifier for this IA_PA; the IAID must be unique among the identifiers for all of this host's IA_PAs. The number space for IA_PA IAIDs is separate from the number spaces for IA_TA and IA_NA IAIDs

T1: The time at which the host should contact the DHCPv6 server from which the prefixes in the IA_PA were obtained to extend the lifetimes of the prefixes assigned to the IA_PA; T1 is a time duration relative to the current time expressed in units of seconds.

T2: The time at which the host should contact any available DHCPv6 server to extend the lifetimes of the prefixes assigned to the IA_PA; T2 is a time duration relative to the current time expressed in units of seconds.

IA_PA-options: Options associated with this IA_PA.

The details of the fields are similar to the IA_PD option description in [RFC3633]. The difference is here a DHCPv6 server and a host involved, while a delegating router and requesting router involved in [RFC3633].

6.2. IA_PA Prefix Option

OPTION_IAPREFIX (26) "IA_PD Prefix Option" defined in Section 10 of [RFC3633] is reused.

Originally, the option is used for conveying prefix information between a delegating router and a requesting router. Here the IA_PD Prefix option is used to specify IPv6 address prefixes associated with an IA_PA in Section 6.1. The IA_PD Prefix option must be encapsulated in the IA_PA-options field of an IA_PA option.

Note, the PD_EXCLUDE option [RFC6603] SHOULD NOT be encapsulated in the IAPREFIX options that are encapsulated in an IA_PA.

7. IANA consideration

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

The OPTION_IA_PA Option (TBA1), described in Section 6.1.

8. Security Considerations

Security considerations in DHCPv6 are described in [RFC3315].

To guard against attacks through prefix assignment, a host and a DHCPv6 server SHOULD use DHCPv6 authentication as described in Section 21, "Authentication of DHCP messages" of [RFC3315] or Secure DHCPv6 [I-D.ietf-dhc-secure-dhcpv6] .

9. Acknowledgements

The authors would like to thanks Suresh Krishnan, Ted Lemon, Bing Liu, Andre Kostur, Gaurav Halwasia, Bernie Volz and other members of DHC WG for their valuable comments.

10. References

10.1. Normative References

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

- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.
- [RFC6603] Korhonen, J., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", RFC 6603, May 2012.

10.2. Informative references

- [RFC3314] Wasserman, M., "Recommendations for IPv6 in Third Generation Partnership Project (3GPP) Standards", RFC 3314, September 2002.
- [RFC4968] Madanapalli, S., "Analysis of IPv6 Link Models for 802.16 Based Networks", RFC 4968, August 2007.
- [I-D.ietf-dhc-secure-dhcpv6]
Jiang, S. and S. Shen, "Secure DHCPv6 Using CGAs", draft-ietf-dhc-secure-dhcpv6-07 (work in progress), September 2012.
- [I-D.ietf-dhc-addr-registration]
Jiang, S., Chen, G., and S. Krishnan, "A Generic IPv6 Addresses Registration Solution Using DHCPv6", draft-ietf-dhc-addr-registration-01 (work in progress), October 2012.

[EUI-64] "Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority", <http://standards.ieee.org/regauth/oui/tutorials/EUI64.html>", March 1997.

Authors' Addresses

Sheng Jiang (editor)
Huawei Technologies
Q14, Huawei Campus, No.156, BeiQing Road
Hai-Dian District, Beijing 100095
P.R. China

Email: jiangsheng@huawei.com

Frank Xia
Huawei Technologies
1700 Alma Dr. Suite 500
Plano, TX 75075

Email: xiayangsong@huawei.com

Behcet Sarikaya
Huawei Technologies
1700 Alma Dr. Suite 500
Plano, TX 75075

Email: sarikaya@ieee.org

Dynamic Host Configuration Working Group
Internet-Draft
Updates: 3315 (if approved)
Intended status: Best Current Practice
Expires: July 11, 2014

D. Hankins
Google
T. Mrugalski
M. Siodelski
ISC
S. Jiang
Huawei Technologies Co., Ltd
S. Krishnan
Ericsson
January 7, 2014

Guidelines for Creating New DHCPv6 Options
draft-ietf-dhc-option-guidelines-17

Abstract

This document provides guidance to prospective DHCPv6 Option developers to help them creating option formats that are easily adoptable by existing DHCPv6 software. It also provides guidelines for expert reviewers to evaluate new registrations. This document updates RFC3315.

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 July 11, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements Language	3
2. Introduction	3
3. When to Use DHCPv6	4
4. General Principles	4
5. Reusing Other Options Formats	5
5.1. Option with IPv6 addresses	6
5.2. Option with a single flag (boolean)	7
5.3. Option with IPv6 prefix	7
5.4. Option with 32-bit integer value	8
5.5. Option with 16-bit integer value	9
5.6. Option with 8-bit integer value	9
5.7. Option with URI	9
5.8. Option with Text String	11
5.9. Option with variable length data	12
5.10. Option with DNS Wire Format Domain Name List	12
6. Avoid Conditional Formatting	13
7. Avoid Aliasing	13
8. Choosing between FQDN and address	14
9. Encapsulated options in DHCPv6	17
10. Additional States Considered Harmful	18
11. Configuration changes occur at fixed times	19
12. Multiple provisioning domains	20
13. Chartering Requirements and Advice for Responsible Area Directors	20
14. Considerations for Creating New Formats	21
15. Option Size	22
16. Singleton options	22
17. Option Order	23
18. Relay Options	24
19. Clients Request their Options	24
20. Transition Technologies	25
21. Recommended sections in the new document	25
21.1. DHCPv6 Client Behavior Text	26
21.2. DHCPv6 Server Behavior Text	27
21.3. DHCPv6 Relay Agent Behavior Text	27
22. Should the new document update existing RFCs?	27
23. Security Considerations	28
24. Privacy considerations	29
25. IANA Considerations	29

26. Acknowledgements	30
27. References	30
27.1. Normative References	30
27.2. Informative References	30
Authors' Addresses	33

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. Introduction

Most protocol developers ask themselves if a protocol will work, or work efficiently. These are important questions, but another less frequently considered question is whether the proposed protocol presents itself needless barriers to adoption by deployed software.

DHCPv6 [RFC3315] software implementors are not merely faced with the task of handling a given option's format on the wire. The option must fit into every stage of the system's process, starting with the user interface used to enter the configuration up to the machine interfaces where configuration is ultimately consumed.

Another frequently overlooked aspect of rapid adoption is whether the option requires operators to be intimately familiar with the option's internal format in order to use it? Most DHCPv6 software provides a facility for handling unknown options at the time of publication. The handling of such options usually needs to be manually configured by the operator. But if doing so requires extensive reading (more than can be covered in a simple FAQ for example), it inhibits adoption.

So although a given solution would work, and might even be space, time, or aesthetically optimal, a given option is presented with a series of ever-worsening challenges to be adopted:

- o If it doesn't fit neatly into existing config files.
- o If it requires source code changes to be adopted, and hence upgrades of deployed software.
- o If it does not share its deployment fate in a general manner with other options, standing alone in requiring code changes or reworking configuration file syntaxes.

- o If the option would work well in the particular deployment environment the proponents currently envision, but has equally valid uses in some other environment where the proposed option format would fail or would produce inconsistent results.

There are many things DHCPv6 option creators can do to avoid the pitfalls in this list entirely, or failing that, to make software implementors lives easier and improve its chances for widespread adoption.

This document is envisaged as a help for protocol developers that define new options and for expert reviewers that review submitted proposals.

3. When to Use DHCPv6

Principally, DHCPv6 carries configuration parameters for its clients. Any knob, dial, slider, or checkbox on the client system, such as "my domain name servers", "my hostname", or even "my shutdown temperature" are candidates for being configured by DHCPv6.

The presence of such a knob isn't enough, because DHCPv6 also presents the extension of an administrative domain - the operator of the network to which the client is currently attached. Someone runs not only the local switching network infrastructure that the client is directly (or wirelessly) attached to, but the various methods of accessing the external Internet via local assist services that the network must also provide (such as domain name servers, or routers). This means that, even if a configuration parameter can potentially be delivered by DHCPv6, it is necessary to evaluate whether it is reasonable for this parameter to be under the control of the administrator of whatever network a client is attached to at any given time.

Note that the client is not required to configure any of these values received via DHCPv6 (e.g., due to having these values locally configured by its own administrator). But it needs to be noted that overriding DHCPv6-provided values may cause the client to be denied certain services in the network to which it has attached. The possibility of having higher level of control over client node configuration is one of the reasons that DHCPv6 is preferred in enterprise networks.

4. General Principles

The primary guiding principle to follow in order to enhance an option's adoptability is reuse. The option should be created in such a way that does not require any new or special case software to

support. If old software currently deployed and in the field can adopt the option through supplied configuration facilities then it's fairly certain that new software can easily formally adopt it.

There are at least two classes of DHCPv6 options: simple options which are provided explicitly to carry data from one side of the DHCPv6 exchange to the other (such as nameservers, domain names, or time servers), and a protocol class of options which require special processing on the part of the DHCPv6 software or are used during special processing (such as the Fully Qualified Domain Name (FQDN) option [RFC4704]), and so forth; these options carry data that is the result of a routine in some DHCPv6 software.

The guidelines laid out here should be applied in a relaxed manner for the protocol class of options. Wherever special case code is already required to adopt the DHCPv6 option, it is substantially more reasonable to format the option in a less generic fashion, if there are measurable benefits to doing so.

5. Reusing Other Options Formats

The easiest approach to manufacturing trivially deployable DHCPv6 Options is to assemble the option out of whatever common fragments fit - possibly allowing a group of data elements to repeat to fill the remaining space (if present) and so provide multiple values. Place all fixed size values at the start of the option, and any variable/indeterminate sized value at the tail end of the option.

This means that implementations will likely be able to reuse code paths designed to support the other options.

There is a tradeoff between the adoptability of previously defined option formats, and the advantages that new or specialized formats can provide. In general, it is usually preferable to reuse previously used option formats.

However, it isn't very practical to consider the bulk of DHCPv6 options already allocated, and consider which of those solve a similar problem. So, the following list of common option format data elements is provided as a shorthand. Please note that it is not complete in terms of exemplifying every option format ever devised.

If more complex options are needed, those basic formats mentioned here may be considered as primitives (or 'fragment types') that can be used to build more complex formats. It should be noted that it is often easier to implement two options with trivial formats than one option with more complex format. That is not unconditional requirement though. In some cases splitting one complex option into

two or more simple options introduces inter-option dependencies that should be avoided. In such a case, it is usually better to keep one complex option.

5.1. Option with IPv6 addresses

This option format is used to carry one or many IPv6 addresses. In some cases the number of allowed address is limited (e.g. to one):

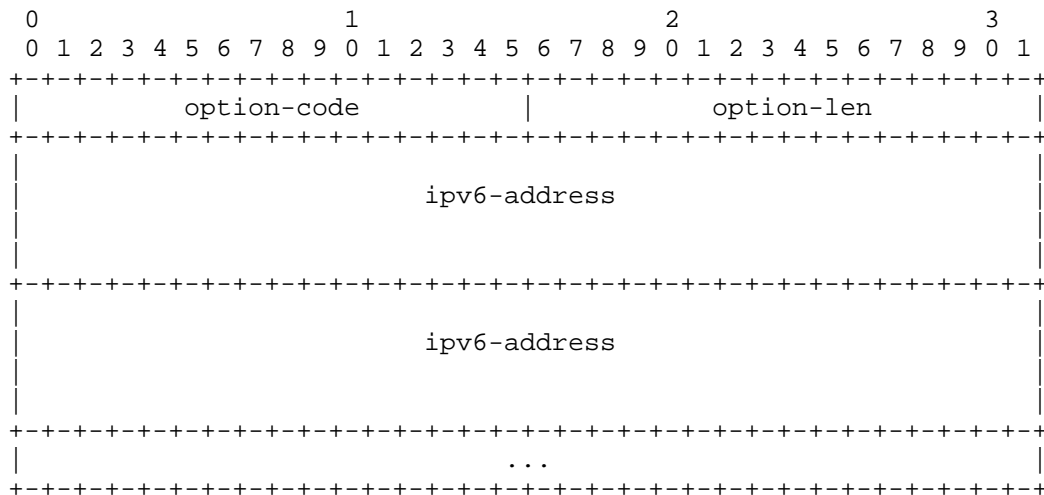


Figure 1: Option with IPv6 address

Examples of use:

- o DHCPv6 server unicast address (a single address only) [RFC3315]
- o SIP Servers IPv6 Address List [RFC3319]
- o DNS Recursive Name Server [RFC3646]
- o NIS Servers [RFC3898]
- o SNTP Servers [RFC4075]
- o Broadcast and Multicast Service Controller IPv6 Address Option for DHCPv6 [RFC4280]
- o MIPv6 Home Agent Address [RFC6610] (a single address only)
- o NTP server [RFC5908] (a single address only)

- o NTP Multicast address [RFC5908] (a single address only)

5.2. Option with a single flag (boolean)

Sometimes it is useful to convey a single flag that can take either on or off values. Instead of specifying an option with one bit of usable data and 7 bits of padding, it is better to define an option without any content. It is the presence or absence of the option that conveys the value. This approach has the additional benefit of absent option designating the default, i.e. administrator has to take explicit actions to deploy the opposite of the default value.

The absence of the option represents the default value and the presence of the option represents the other value, but that does not necessarily mean that absence is "off" (or "false") and presence is "on" (or "true"). That is, if it's desired that the default value for a bistable option is "true"/"on", then the presence of that option would turn it off (make it false). If the option presence signifies off/false state, that should be reflected in the option name, e.g. OPTION_DISABLE_FOO.

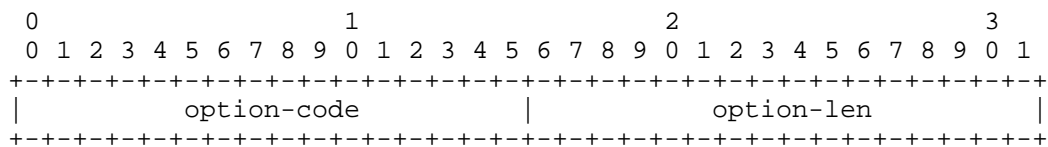


Figure 2: Option for conveying boolean

Examples of use:

- o DHCPv6 rapid-commit [RFC3315]

5.3. Option with IPv6 prefix

Sometimes there is a need to convey an IPv6 prefix. The information to be carried by such an option includes the 128-bit IPv6 prefix together with a length of this prefix taking values from 0 to 128. Using the simplest approach, the option could convey this data in two fixed length fields: one carrying prefix length, another carrying the prefix. However, in many cases /64 or shorter prefixes are used. This implies that the large part of the prefix data carried by the option would have its bits set to zero and would be unused. In order to avoid carrying unused data, it is recommended to store prefix in the variable length data field. The appropriate option format is defined as follows:

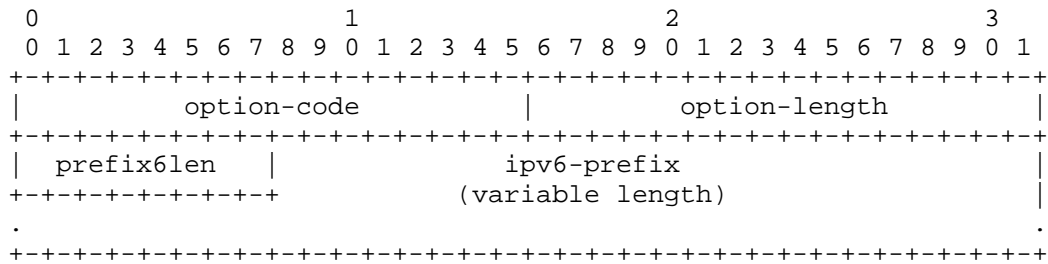


Figure 3: Option with IPv6 Prefix

option-length is set to 1 + length of the IPv6 prefix.

prefix6len is one octet long and specifies the length in bits of the IPv6 prefix. Typically allowed values are 0 to 128.

ipv6-prefix field is a variable length field that specifies the IPv6 prefix. The length is $(\text{prefix6len} + 7) / 8$. This field is padded with zero bits up to the nearest octet boundary when prefix6len is not divisible by 8.

Examples of use:

- o Default Mapping Rule [I-D.ietf-softwire-map-dhcp]

For example, the prefix 2001:db8::/60 would be encoded with an option-length of 9, prefix6-len would be set to 60, the ipv6-prefix would be 8 octets and would contain octets 20 01 0d b8 00 00 00 00.

It should be noted that the IAPREFIX option defined by [RFC3633] uses a full length 16-octet prefix field. The concern about option length was not well understood at the time of its publication.

5.4. Option with 32-bit integer value

This option format can be used to carry 32 bit-signed or unsigned integer value:

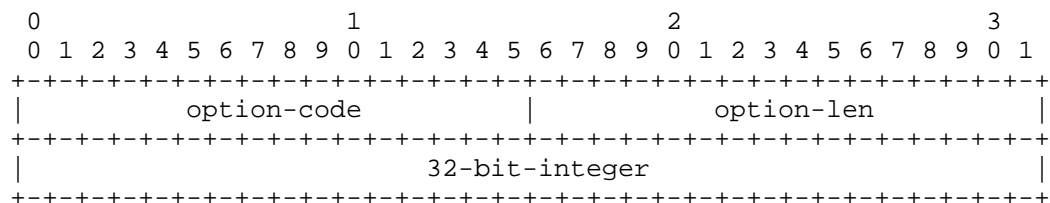


Figure 4: Option with 32-bit-integer value

Examples of use:

- o Information Refresh Time [RFC4242]

5.5. Option with 16-bit integer value

This option format can be used to carry 16-bit signed or unsigned integer values:

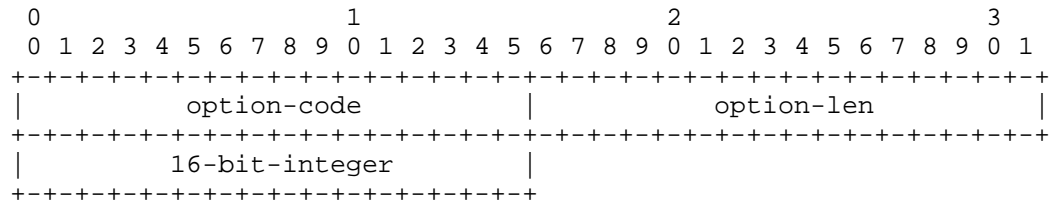


Figure 5: Option with 16-bit integer value

Examples of use:

- o Elapsed Time [RFC3315]

5.6. Option with 8-bit integer value

This option format can be used to carry 8-bit integer values:

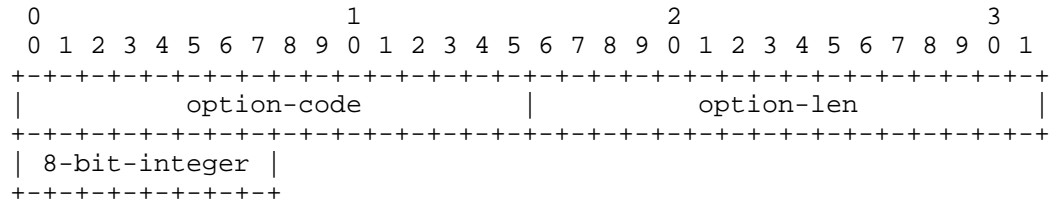


Figure 6: Option with 8-bit integer value

Examples of use:

- o DHCPv6 Preference [RFC3315]

5.7. Option with URI

A Uniform Resource Identifier (URI) [RFC3986] is a compact sequence of characters that identifies an abstract or physical resource. The term "Uniform Resource Locator" (URL) refers to the subset of URIs that, in addition to identifying a resource, provide a means of locating the resource by describing its primary access mechanism

(e.g., its network "location"). This option format can be used to carry a single URI:

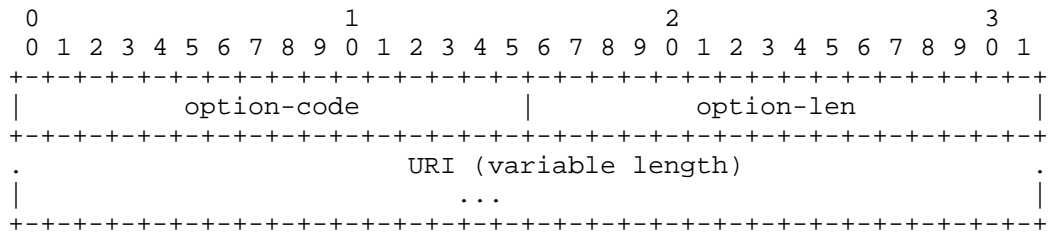


Figure 7: Option with URI

Examples of use:

- o Boot File URL [RFC5970]

An alternate encoding to support multiple URIs is available. An option must be defined to use either the single URI format above or the multiple URI format below depending on whether a single is always sufficient or if multiple URIs are possible.

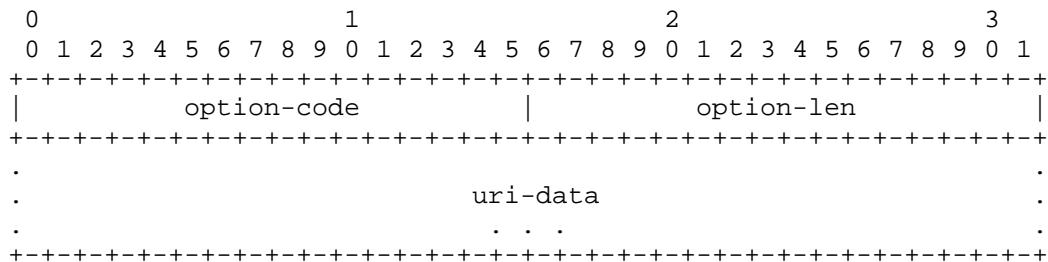
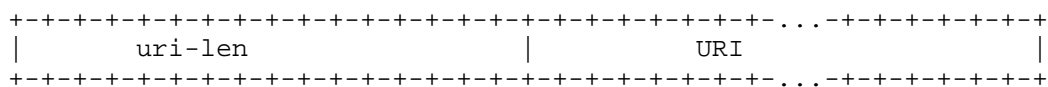


Figure 8: Option with multiple URIs

Each instance of the uri-data is formatted as follows:



The uri-len is two octets long and specifies the length of the uri data. Although URI format in theory supports up to 64k of data, in practice large chunks of data may be problematic. See Section 15 for details.

5.8. Option with Text String

A text string is a sequence of characters that have no semantics. The encoding of the text string **MUST** be specified. Unless otherwise specified, all text strings in newly defined options are expected to be Unicode strings that are encoded using UTF-8 [RFC3629] in Net-Unicode form [RFC5198]. Please note that all strings containing only 7 bit ASCII characters are also valid UTF-8 Net-Unicode strings.

If a data format has semantics other than just being text, it is not a string. E.g., a FQDN is not a string, and a URI is also not a string, because they have different semantics. A string must not include any terminator (such as a null byte). The null byte is treated as any other character and does not have any special meaning. This option format can be used to carry a text string:

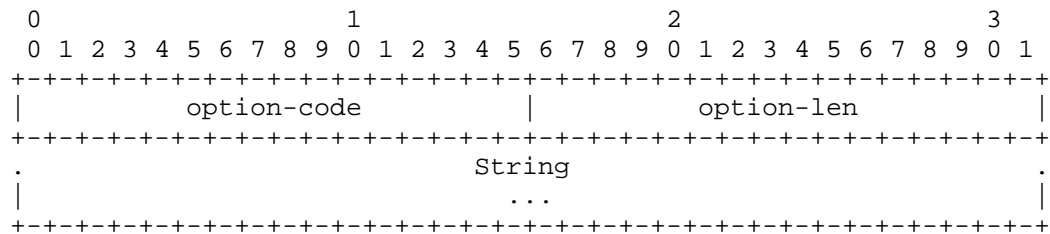


Figure 9: Option with text string

Examples of use:

- o Timezone Options for DHCPv6 [RFC4833]

An alternate encoding to support multiple text strings is available. An option must be defined to use either the single text string format above or the multiple text string format below depending on whether a single is always sufficient or if multiple text strings are possible.

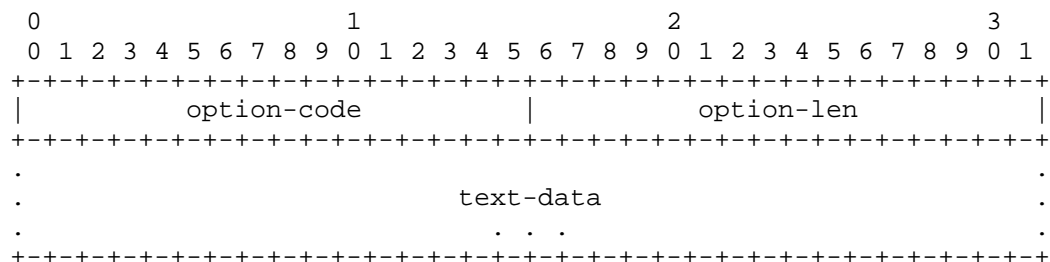


Figure 10: Option with multiple text strings

Each instance of the text-data is formatted as follows:

```
+-----+-----+-----+-----+-----+-----+-----+-----+...+-----+
|           text-len           |           String           |
+-----+-----+-----+-----+-----+-----+-----+-----+...+-----+
```

The text-len is two octets long and specifies the length of the string.

5.9. Option with variable length data

This option can be used to carry variable length data of any kind. Internal representation of carried data is option specific. Whenever this format is used by the new option being defined, the data encoding should be documented.

This option format provides a lot of flexibility to pass data of almost any kind. Though, whenever possible it is highly recommended to use more specialized options, with field types better matching carried data types.

```

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 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+...+-----+
|           option-code           |           option-len           |
+-----+-----+-----+-----+-----+-----+-----+-----+...+-----+
.
.           variable length data           .
.
+-----+-----+-----+-----+-----+-----+-----+-----+...+-----+
```

Figure 11: Option with variable length data

Examples of use:

- o Client Identifier [RFC3315]
- o Server Identifier [RFC3315]

5.10. Option with DNS Wire Format Domain Name List

This option is used to carry 'domain search' lists or any host or domain name. It uses the same format as described in Section 5.9, but with the special data encoding, described in section 8 of [RFC3315]. This data encoding supports carrying multiple instances of hosts or domain names in a single option, by terminating each instance with the byte value of 0.

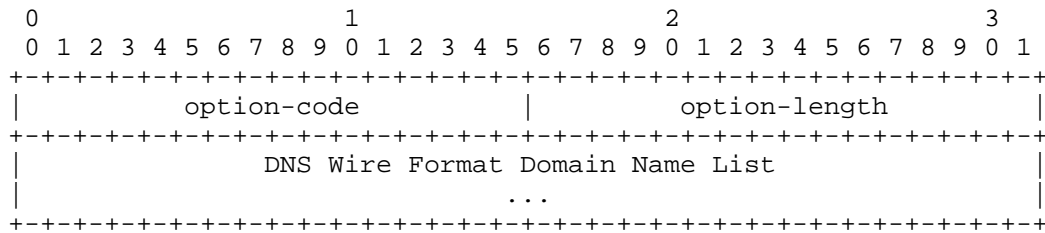


Figure 12: Option with DNS Wire Format Domain Name List

Examples of use:

- o SIP Servers Domain Name List [RFC3319] (many domains)
- o NIS Domain Name (many domains) [RFC3898] (many domains)
- o LoST Server Domain name [RFC5223]
- o LIS Domain name [RFC5986]
- o DS-Lite AFTR location [RFC6334] (a single FQDN)
- o Home Network Identifier [RFC6610] (a single FQDN)
- o Home Agent FQDN [RFC6610] (a single FQDN)

6. Avoid Conditional Formatting

Placing an octet at the start of the option which informs the software how to process the remaining octets of the option may appear simple to the casual observer. But the only conditional formatting methods that are in widespread use today are 'protocol' class options. Therefore conditional formatting requires new code to be written and complicates future interoperability should new conditional formats be added; and existing code has to ignore conditional format that it does not support.

7. Avoid Aliasing

Options are said to be aliases of each other if they provide input to the same configuration parameter. A commonly proposed example is to configure the location of some new service ("my foo server") using a binary IP address, a domain name field, and an URL. This kind of aliasing is undesirable, and is not recommended.

In this case, where three different formats are supposed, it more than triples the work of the software involved, requiring support for

not merely one format, but support to produce and digest all three. Furthermore, code development and testing must cover all possible combinations of defined formats. Since clients cannot predict what values the server will provide, they must request all formats. So in the case where the server is configured with all formats, DHCPv6 message bandwidth is wasted on option contents that are redundant. Also, the DHCPv6 option number space is wasted, as three new option codes are required, rather than one.

It also becomes unclear which types of values are mandatory, and how configuring some of the options may influence the others. For example, if an operator configures the URL only, should the server synthesize a domain name and IP address?

A single configuration value on a host is probably presented to the operator (or other software on the machine) in a single field or channel. If that channel has a natural format, then any alternative formats merely make more work for intervening software in providing conversions.

So the best advice is to choose the one method that best fulfills the requirements, be that for simplicity (such as with an IP address and port pair), late binding (such as with DNS), or completeness (such as with a URL).

8. Choosing between FQDN and address

Some parameters may be specified as FQDN or an address. In most cases one or the other should be used. This section discusses pros and cons of each approach and is intended to help make an informed decision in that regard. It is strongly discouraged to define both option types at the same time (see Section 7), unless there is sufficient motivation to do so.

There is no single recommendation that works for every case. It very much depends on the nature of the parameter being configured. For parameters that are network specific or represent certain aspects of network infrastructure, like available mobility services, in most cases addresses are a more usable choice. For parameters that can be considered application specific configuration, like SIP servers, it is usually better to use FQDN.

Applications are often better suited to deal with FQDN failures than with address failures. Most operating systems provide a way to retry FQDN resolution if the previous attempt fails. That type of error recovery is supported by a great number of applications. On the other hand, there is typically no API available for applications to reconfigure over DHCP to get a new address value if the one received

is no longer appropriate. This problem may be usually addressed by providing a list of addresses, rather than just a single one. That, on the other hand, requires defined procedure how multiple addresses should be used (all at once, round robin, try first and fail over to the next if it fails etc.).

FQDN provide a higher level of indirection and ambiguity. In many cases that may be considered a benefit, but can be considered a flaw in others. For example, one operator suggested to have the same name being resolved to different addresses depending on the point of attachment of the host doing resolution. This is one way to provide localized addressing. However, in order to do this, it is necessary to violate the DNS convention that a query on a particular name should always return the same answer (aside from ordering of IP addresses in the response, which is supposed to be varied by the name server). This same locality of reference for configuration information can be achieved directly using DHCP, since the DHCP server must know the network topology in order to provide IP address or prefix configuration.

The other type of ambiguity is related to multiple provisioning domains (see Section 12). The stub resolver on the DHCP client cannot at present be assumed to make the DNS query for a DHCP-supplied FQDN on the same interface on which it received its DHCP configuration, and may therefore get a different answer from the DNS than was intended.

This is particularly a problem when the normal expected use of the option makes sense with private DNS zone(s), as might be the case on an enterprise network. It may also be the case that the client has an explicit DNS server configured, and may therefore never query the enterprise network's internal DNS server.

FQDN does require a resolution into an actual address. This implies the question when the FQDN resolution should be conducted. There are a couple of possible answers: a) by the server, when it is started, b) by the server, when it is about to send an option, c) by the client, immediately after receiving an option, d) by the client, when the content of the option is actually consumed. For a), b) and possibly c), the option should really convey an address, not FQDN. The only real incentive to use FQDN is case d). It is the only case that allows possible changes in the DNS to be picked up by clients.

If the parameter is expected to be used by constrained devices (low power, battery operated, low capabilities) or in very lossy networks, it may be appealing to drop the requirement of having DNS resolution being performed and use addresses. Another example of a constrained device is a network booted device, where despite the fact that the

node itself is very capable once it's booted, the boot prom is quite constrained.

Another aspect that should be considered is time required for the clients to notice any configuration changes. Consider a case where a server configures a service A using address and service B using FQDN. When an administrator decides to update the configuration, he or she can update the DHCP server configuration to change both services. If the clients do not support reconfigure (which is an optional feature of RFC3315, but in some environments, e.g. cable modems, is mandatory), the configuration will be updated on clients after T1 timer elapses. Depending on the nature of the change (is it a new server added to a cluster of already operating servers or a new server that replaces the only available server that crashed?), this may be an issue. On the other hand, updating service B may be achieved with DNS record update. That information may be cached by caching DNS servers for up to TTL. Depending on the values of T1 and TTL, one update may be faster than another. Furthermore, depending on the nature of the change (planned modification or unexpected failure), T1 or TTL may be lowered before the change to speed up new configuration adoption.

Simply speaking protocol designers don't know what the TTL or the T1 time will be, so they can't make assumptions about whether a DHCP option will be refreshed more quickly based on T1 or TTL.

Addresses have a benefit of being easier to implement and handle by the DHCP software. An address option is simpler to use, its validation is trivial (multiple of 16 constitutes a valid option), is explicit and does not allow any ambiguity. It is faster (does not require extra round trip time), so it is more efficient, which can be especially important for energy restricted devices. It also does not require that the client implements DNS resolution.

FQDN imposes a number of additional failure modes and issues that should be dealt with:

1. The client must have a knowledge about available DNS servers. That typically means that option DNS_SERVERS [RFC3646] is mandatory. This should be mentioned in the draft that defines new option. It is possible that the server will return FQDN option, but not the DNS Servers option. There should be a brief discussion about it;
2. The DNS may not be reachable;
3. DNS may be available, but may not have appropriate information (e.g. no AAAA records for specified FQDN);

4. Address family must be specified (A, AAAA or any); the information being configured may require specific address family (e.g. IPv6), but there may be a DNS record only of another type (e.g. A only with IPv4 address).
5. What should the client do if there are multiple records available (use only the first one, use all, use one and switch to the second if the first fails for whatever reason, etc.); This may be an issue if there is an expectation that the parameter being configured will need exactly one address;
6. Multi-homed devices may be connected to different administrative domains with each domain providing different information in DNS (e.g. an enterprise network exposing private domains). Client may send DNS queries to a different DNS server;
7. It should be mentioned if Internationalized Domain Names are allowed. If they are, DNS option encoding should be specified.

Address options that are used with overly long T1 (renew timer) values have some characteristics of hardcoded values. That is strongly discouraged. See [RFC4085] for an in depth discussion. If the option may appear in Information-Request, its lifetime should be controlled using information refresh time option [RFC4242].

One specific case that makes the choice between address and FQDN not obvious is a DNSSEC bootstrap scenario. DNSSEC validation imposes a requirement for clock sync (to the accuracy reasonably required to consider signature inception and expiry times). This often implies usage of NTP configuration. However, if the NTP is provided as FQDN, there is no way to validate its DNSSEC signature. This is somewhat weak argument though, as providing NTP server as an address is also not verifiable using DNSSEC. If the trustworthiness of the configuration provided by DHCP server is in question, DHCPv6 offers authentication mechanisms that allow server authentication.

9. Encapsulated options in DHCPv6

Most options are conveyed in a DHCPv6 message directly. Although there is no codified normative language for such options, they are often referred to as top-level options. Many options may include other options. Such inner options are often referred to as encapsulated or nested options. Those options are sometimes called sub-options, but this term actually means something else, and therefore should never be used to describe encapsulated options. It is recommended to use term "encapsulated" as this terminology is used in [RFC3315]. The difference between encapsulated and sub-options are that the former uses normal DHCPv6 option numbers, while the

latter uses option number space specific to a given parent option. It should be noted that, contrary to DHCPv4, there is no shortage of option numbers. Therefore almost all options share a common option space. For example option type 1 meant different things in DHCPv4, depending if it was located in top-level or inside of Relay Agent Information option. There is no such ambiguity in DHCPv6 (with the exception of [RFC5908], which SHOULD NOT be used as a template for future DHCP option definitions).

From the implementation perspective, it is easier to implement encapsulated options rather than sub-options, as the implementers do not have to deal with separate option spaces and can use the same buffer parser in several places throughout the code.

Such encapsulation is not limited to one level. There is at least one defined option that is encapsulated twice: Identity Association for Prefix Delegation (IA_PD, defined in [RFC3633], section 9) conveys IA Prefix (IAPREFIX, defined in [RFC3633], section 10). Such delegated prefix may contain an excluded prefix range that is represented by PD_EXCLUDE option that is conveyed as encapsulated inside IAPREFIX (PD_EXCLUDE, defined in [RFC6603]). It seems awkward to refer to such options as sub-sub-option or doubly encapsulated option, therefore "encapsulated option" term is typically used, regardless of the nesting level.

When defining a DHCP-based configuration mechanism for a protocol that requires something more complex than a single option, it may be tempting to group configuration values using sub-options. That should preferably be avoided, as it increases complexity of the parser. It is much easier, faster and less error prone to parse a large number of options on a single (top-level) scope, than parse options on several scopes. The use of sub-options should be avoided as much as possible, but it is better to use sub-options rather than conditional formatting.

It should be noted that currently there is no clear way defined for requesting sub-options. Most known implementations are simply using top-level ORO for requesting both top-level options and encapsulated options.

10. Additional States Considered Harmful

DHCP is a protocol designed for provisioning clients. Less experienced protocol designers often assume that it is easy to define an option that will convey a different parameter for each client in a network. Such problems arose during designs of MAP [I-D.ietf-software-map-dhcp] and 4rd [I-D.ietf-software-4rd]. While it would be easier for provisioned clients to get ready to use per-

client option values, such requirement puts exceedingly large loads on the server side. The new extensions may introduce new implementation complexity and additional database state on the server. Alternatives should be considered, if possible. As an example, [I-D.ietf-softwire-map-dhcp] was designed in a way that all clients are provisioned with the same set of MAP options and each provisioned client uses its unique address and delegated prefix to generate client-specific information. Such a solution does not introduce any additional state for the server and therefore scales better.

It also should be noted that contrary to DHCPv4, DHCPv6 keeps several timers for renewals. Each IA_NA (addresses) and IA_PD (prefixes) contains T1 and T2 timers that designate time after which client will initiate renewal. Those timers apply only to its own IA containers. Refreshing other parameters should be initiated after a time specified in the Information Refresh Time Option (defined in [RFC4242]), carried in the Reply message and returned in response to Information-Request message. Introducing additional timers make deployment unnecessarily complex and SHOULD be avoided.

11. Configuration changes occur at fixed times

In general, DHCPv6 clients only refresh configuration data from the DHCP server when the T1 timer expires. Although there is a RECONFIGURE mechanism that allows a DHCP server to request that clients initiate reconfiguration, support for this mechanism is optional and cannot be relied upon.

Even when DHCP clients refresh their configuration information, not all consumers of DHCP-sourced configuration data notice these changes. For instance, if a server is started using parameters received in an early DHCP transaction, but does not check for updates from DHCP, it may well continue to use the same parameter indefinitely. There are a few operating systems that take care of reconfiguring services when the client moves to a new network (e.g. based on mechanisms like [RFC4436], [RFC4957] or [RFC6059]), but it's worth bearing in mind that a renew may not always result in the client taking up new configuration information that it receives.

In light of the above, when designing an option you should take into consideration the fact that your option may hold stale data that will only be updated at an arbitrary time in the future.

12. Multiple provisioning domains

In some cases there could be more than one DHCPv6 server on a link, with each providing a different set of parameters. One notable example of such a case is a home network with a connection to two independent ISPs.

The DHCPv6 protocol specification does not provide clear advice on how to handle multiple provisioning sources. Although [RFC3315] states that a client that receives more than one ADVERTISE message, may respond to one or more of them, such capability has not been observed in existing implementations. Existing clients will pick one server and will continue configuration process with that server, ignoring all other servers.

In addition, a node that acts as a DHCPv6 client may be connected to more than one physical network. In this case, it will in most cases operate a separate DHCP client state machine on each interface, acquiring different, possibly conflicting information through each. This information will not be acquired in any synchronized way.

Existing nodes cannot be assumed to systematically segregate configuration information on the basis of its source; as a result, it is quite possible that a node may receive an FQDN on one network interface, but do the DNS resolution on a different network interface, using different DNS servers. As a consequence, DNS resolution done by the DHCP server is more likely to behave predictably than DNS resolution done on a multi-interface or multi-homed client.

This is a generic DHCP protocol issue and should not be dealt within each option separately. This issue is better dealt with using a protocol-level solution and fixing this problem should not be attempted on a per option basis. Work is ongoing in the IETF to provide a systematic solution to this problem.

13. Chartering Requirements and Advice for Responsible Area Directors

Adding a simple DHCP option is straightforward, and generally something that any working group can do, perhaps with some help from designated DHCP experts. However, when new fragment types need to be devised, this requires the attention of DHCP experts, and should not be done in a working group that doesn't have a quorum of such experts. This is true whether the new fragment type has the same structure as an existing fragment type but has different semantics, or the new format has a new structure.

Responsible Area Directors for working groups that wish to add a work item to a working group charter to define a new DHCP option should get clarity from the working group as to whether the new option will require a new fragment type or new semantics, or whether it is a simple DHCP option that fits existing definitions.

If a working group needs a new fragment type, it is preferable to see if another working group exists whose members already have sufficient expertise to evaluate the new work. If such a working group is available, the work should be chartered in that working group instead. If there is no other working group with DHCP expertise that can define the new fragment type, the responsible AD should seek help from known DHCP experts within the IETF to provide advice and frequent early review as the original working group defines the new fragment type.

In either case, the new option should be defined in a separate document, and the work should focus on defining a new format that generalizes well and can be reused, rather than a single-use fragment type. The working group that needs the new fragment type can define their new option referencing the new fragment type document, and the work can generally be done in parallel, avoiding unnecessary delays. Having the definition in its own document will foster reuse of the new fragment type.

The responsible AD should work with all relevant working group chairs and DHCP experts to ensure that the new fragment type document has in fact been carefully reviewed by the experts and appears satisfactory.

Responsible area directors for working groups that are considering defining options that actually update the DHCP protocol, as opposed to simple options, should go through a process similar to that described above when trying to determine where to do the work. Under no circumstances should a working group be given a charter deliverable to define a new DHCP option, and then on the basis of that charter item actually make updates to the DHCP protocol.

14. Considerations for Creating New Formats

When defining new options, one specific consideration to evaluate is whether or not options of a similar format would need to have multiple or single values encoded (whatever differs from the current option), and how that might be accomplished in a similar format.

When defining a new option, it is best to synthesize the option format using fragment types already in use. However, in some cases there may be no fragment type that accomplishes the intended purpose.

The matter of size considerations and option order are further discussed in Section 15 and Section 17.

15. Option Size

DHCPv6 [RFC3315] allows for packet sizes up to 64KB. First, through its use of link-local addresses, it avoids many of the deployment problems that plague DHCPv4, and is actually an UDP over IPv6 based protocol (compared to DHCPv4, which is mostly UDP over IPv4 protocol, but with layer 2 hacks). Second, RFC 3315 explicitly refers readers to RFC 2460 Section 5, which describes an MTU of 1280 octets and a minimum fragment reassembly of 1500 octets. It's feasible to suggest that DHCPv6 is capable of having larger options deployed over it, and at least no common upper limit is yet known to have been encoded by its implementors. It is not really possible to describe a fixed limit that cleanly divides workable option sizes from those that are too big.

It is advantageous to prefer option formats which contain the desired information in the smallest form factor that satisfies the requirements. Common sense still applies here. It is better to split distinct values into separate octets rather than propose overly complex bit shifting operations to save several bits (or even an octet or two) that would be padded to the next octet boundary anyway.

DHCPv6 does allow for multiple instances of a given option, and they are treated as distinct values following the defined format, however this feature is generally preferred to be restricted to protocol class features (such as the IA_* series of options). In such cases, it is better to define an option as an array if it is possible. It is recommended to clarify (with normative language) whether a given DHCPv6 option may appear once or multiple times. The default assumption is only once.

In general, if a lot of data needs to be configured (for example, some option lengths are quite large), DHCPv6 may not be the best choice to deliver such configuration information and SHOULD simply be used to deliver a URI that specifies where to obtain the actual configuration information.

16. Singleton options

Although [RFC3315] states that each option type MAY appear more than once, the original idea was that multiple instances are reserved for stateful options, like IA_NA or IA_PD. For most other options it is usually expected that they will appear at most once. Such options are called singleton options. Sadly, RFCs have often failed to

clearly specify whether a given option can appear more than once or not.

Documents that define new options SHOULD state whether these options are singletons or not. Unless otherwise specified, newly defined options are considered to be singletons. If multiple instances are allowed, the document MUST explain how to use them. Care should be taken to not assume they will be processed in the order they appear in the message. See Section 17 for more details.

When deciding whether a single or multiple option instances are allowed in a message, take into consideration how the content of the option will be used. Depending on the service being configured it may or may not make sense to have multiple values configured. If multiple values make sense, it is better to explicitly allow that by using option format that allows multiple values within one option instance.

Allowing multiple option instances often leads to confusion. Consider the following example. Basic DS-Lite architecture assumes that the B4 element (DHCPv6 client) will receive AFTR option and establish a single tunnel to configured tunnel termination point (AFTR). During standardization process of [RFC6334] there was a discussion whether multiple instances of DS-Lite tunnel option should be allowed. This created an unfounded expectation that the clients receiving multiple instances of the option will somehow know when one tunnel endpoint goes off-line and do some sort of failover between other values provided in other instances of the AFTR option. Others assumed that if there are multiple options, the client will somehow do a load balancing between provided tunnel endpoints. Neither failover nor load balancing was defined for DS-Lite architecture, so it caused confusion. It was eventually decided to allow only one instance of the AFTR option.

17. Option Order

Option order, either the order among many DHCPv6 options or the order of multiple instances of the same option, SHOULD NOT be significant. New documents MUST NOT assume any specific option processing order.

As there is no explicit order for multiple instances of the same option, an option definition SHOULD instead restrict ordering by using a single option that contains ordered fields.

As [RFC3315] does not impose option order, some implementations use hash tables to store received options (which is a conformant behavior). Depending on the hash implementation, the processing

order is almost always different then the order in which options appeared in the packet on wire.

18. Relay Options

In DHCPv4, all relay options are organized as sub-options within DHCP Relay Agent Information Option[RFC3046]. And an independent number space called "DHCP Relay Agent Sub-options" is maintained by IANA. Different from DHCPv4, in DHCPv6, Relay options are defined in the same way as client/server options, and they too use the same option number space as client/server options. Future DHCPv6 Relay options MUST be allocated from this single DHCPv6 Option number space.

E.g. the Relay-Supplied Options Option [RFC6422] may also contain some DHCPv6 options as permitted, such as the EAP Re-authentication Protocol (ERP) Local Domain Name DHCPv6 Option [RFC6440].

19. Clients Request their Options

The DHCPv6 Option Request Option (OPTION_ORO) [RFC3315], is an option that serves two purposes - to inform the server what options the client supports and to inform what options the client is willing to consume.

For some options, such as the options required for the functioning of the DHCPv6 protocol itself, it doesn't make sense to require that they be explicitly requested using the Option Request Option. In all other cases, it is prudent to assume that any new option must be present on the relevant option request list if the client desires to receive it.

It is tempting to add text that requires the client to include a new option in Option Request Option list, similar to this text: "Clients MUST place the foo option code on the Option Request Option list, clients MAY include option foo in their packets as hints for the server as values the desire, and servers MUST include option foo when the client requested it (and the server has been so configured)". Such text is discouraged as there are several issues with it. First, it assumes that client implementation that supports a given option will always want to use it. This is not true. The second and more important reason is that such text essentially duplicates mechanism already defined in [RFC3315]. It is better to simply refer to the existing mechanism rather than define it again. See Section 21 for proposed examples on how to do that.

Creators of DHCPv6 options cannot not assume special ordering of options either as they appear in the option request option, or as they appear within the packet. Although it is reasonable to expect

that options will be processed in the order they appear in ORO, server software is not required to sort DHCPv6 options into the same order in reply messages.

It should also be noted that options values are never aligned within the DHCP packet, even the option code and option length may appear on odd byte boundaries.

20. Transition Technologies

Transition from IPv4 to IPv6 is progressing. Many transition technologies are proposed to speed it up. As a natural consequence there are also DHCP options proposed to provision those proposals. The inevitable question is whether the required parameters should be delivered over DHCPv4 or DHCPv6. Authors often don't give much thought about it and simply pick DHCPv6 without realizing the consequences. IPv6 is expected to stay with us for many decades, and so is DHCPv6. There is no mechanism available to deprecate an option in DHCPv6, so any options defined will stay with us as long as DHCPv6 protocol itself. It seems likely that such options defined to transition from IPv4 will outlive IPv4 by many decades. From that perspective it is better to implement provisioning of the transition technologies in DHCPv4, which will be obsoleted together with IPv4.

When the network infrastructure becomes IPv6-only, the support for IPv4-only nodes may still be needed. In such a scenario, a mechanism for providing IPv4 configuration information over IPv6-only networks such as [I-D.ietf-dhc-v4configuration] may be needed.

21. Recommended sections in the new document

There are three major entities in DHCPv6 protocol: server, relay agent, and client. It is very helpful for implementers to include separate sections that describe operation for those three major entities. Even when a given entity does not participate, it is useful to have a very short section stating that it must not send a given option and must ignore it when received.

There is also a separate entity called requestor, which is a special client-like type that participates in leasequery protocol [RFC5007] and [RFC5460]. A similar section for the requestor is not required, unless the new option has anything to do with requestor (or it is likely that the reader may think that is has). It should be noted that while in the majority of deployments, requestor is co-located with relay agent, those are two separate entities from the protocol perspective and they may be used separately. There are stand-alone requestor implementations available.

The following sections include proposed text for such sections. That text is not required to appear, but it is appropriate in most cases. Additional or modified text specific to a given option is often required.

Although requestor is somewhat uncommon functionality, its existence should be noted, especially when allowing or disallowing options to appear in certain message or being sent by certain entities. Additional message types may appear in the future, besides types defined in [RFC3315]. Therefore authors are encouraged to familiarize themselves with a list of currently defined DHCPv6 messages available on IANA website [iana].

Typically new options are requested by clients and assigned by the server, so there is no specific relay behavior. Nevertheless it is good to include a section for relay agent behavior and simply state that there are no additional requirements for relays. The same applies for client behavior if the options are to be exchanged between relay and server.

Sections that contain option definitions MUST include formal verification procedure. Often it is very simple, e.g. option that conveys IPv6 address must be exactly 16 bytes long, but sometimes the rules are more complex. It is recommended to refer to existing documents (e.g. section 8 of RFC3315 for domain name encoding) rather than trying to repeat such rules.

21.1. DHCPv6 Client Behavior Text

Clients MAY request option foo, as defined in [RFC3315], sections 17.1.1, 18.1.1, 18.1.3, 18.1.4, 18.1.5 and 22.7. As a convenience to the reader, we mention here that the client includes requested option codes in Option Request Option.

Optional text (if client's hints make sense): Client also MAY include option foo in its SOLICIT, REQUEST, RENEW, REBIND and INFORMATION-REQUEST messages as a hint for the server regarding preferred option values.

Optional text (if the option contains FQDN): If the client requests an option that conveys an FQDN, it is expected that the contents of that option will be resolved using DNS. Hence the following text may be useful: Clients that request option foo SHOULD also request option OPTION_DNS_SERVERS specified in [RFC3646].

Clients MUST discard option foo if it is invalid (i.e. did not pass validation steps defined in Section X.Y).

Optional text (if option foo is expected to be exchanged between relays and servers): Option foo is exchanged between relays and servers only. Clients are not aware of the usage of option foo. Clients MUST ignore received option foo.

21.2. DHCPv6 Server Behavior Text

Sections 17.2.2 and 18.2 of [RFC3315] govern server operation in regards to option assignment. As a convenience to the reader, we mention here that the server will send option foo only if configured with specific values for foo and the client requested it.

Optional text: Option foo is a singleton. Servers MUST NOT send more than one instance of foo option.

Optional text (if server is never supposed to receive option foo): Servers MUST ignore incoming foo option.

21.3. DHCPv6 Relay Agent Behavior Text

It's never appropriate for a relay agent to add options to a message heading toward the client, and relay agents don't actually construct Relay-Reply messages anyway.

Optional text (if foo option is exchanged between clients and server or between requestors and servers): There are no additional requirements for relays.

Optional text (if relays are expected to insert or consume option foo): Relay agents MAY include option foo in a Relay-Forw when forwarding packets from clients to the servers.

22. Should the new document update existing RFCs?

Authors often ask themselves a question whether their proposal updates exist RFCs, especially 3315. In April 2013 there were about 80 options defined. Had all documents that defined them also updated RFC3315, comprehension of such a document set would be extremely difficult. It should be noted that "extends" and "updates" are two very different verbs. If a new draft defines a new option that clients request and servers provide, it merely extends current standards, so "updates 3315" is not required in the new document header. On the other hand, if a new document replaces or modifies existing behavior, includes clarifications or other corrections, it should be noted that it updates the other document. For example, [RFC6644] clearly updates [RFC3315] as it replaces existing with new text.

If in doubt, authors should try to answer a question whether implementor reading the base RFC alone (without reading the new draft) would be able to properly implement the software. If the base RFC is sufficient, that the new draft most probably does not update the base RFC. On the other hand, if reading your draft is necessary to properly implement the base RFC, then the new draft most likely updates the base RFC.

23. Security Considerations

DHCPv6 does have an Authentication mechanism ([RFC3315]) that makes it possible for DHCPv6 software to discriminate between authentic endpoints and man-in-the-middle. Other authentication mechanisms may optionally be deployed. Sadly, as of late 2013, the authentication in DHCPv6 is rarely used and support for it is not common in existing implementations. Some specific deployment types make it mandatory (or parts of thereof, e.g. DOCSIS3.0 compatible cable modems require reconfigure-key support), so in certain cases specific authentication aspects can be relied upon. That is not true in the generic case, though.

So, while creating a new option, it is prudent to assume that the DHCPv6 packet contents are always transmitted in the clear, and actual production use of the software will probably be vulnerable at least to man-in-the-middle attacks from within the network, even where the network itself is protected from external attacks by firewalls. In particular, some DHCPv6 message exchanges are transmitted to multicast addresses that are likely broadcast anyway.

If an option is of a specific fixed length, it is useful to remind the implementer of the option data's full length. This is easily done by declaring the specific value of the 'length' tag of the option. This helps to gently remind implementers to validate option length before digesting them into likewise fixed length regions of memory or stack.

If an option may be of variable size (such as having indeterminate length fields, such as domain names or text strings), it is advisable to explicitly remind the implementor to be aware of the potential for long options. Either define a reasonable upper limit (and suggest validating it), or explicitly remind the implementor that an option may be exceptionally long (to be prepared to handle errors rather than truncate values).

For some option contents, out of bound values may be used to breach security. An IP address field might be made to carry a loopback address, or local multicast address, and depending on the protocol this may lead to undesirable results. A domain name field may be

filled with contrived contents that exceed the limitations placed upon domain name formatting - as this value is possibly delivered to "internal configuration" records of the system, it may be implicitly trusted without being validated.

Authors of drafts defining new DHCP options are therefore strongly advised to explicitly define validation measures that recipients of such options are required to do before processing such options. However, validation measures already defined by RFC3315 or other specifications referenced by the new option document are redundant, and can introduce errors, so authors are equally strongly advised to refer to the base specification for any such validation language rather than copying it into the new specification.

Also see Section 24.

24. Privacy considerations

As discussed in Section 23 the DHCPv6 packets are typically transmitted in the clear, so they are susceptible to eavesdropping. This should be considered when defining options that may convey personally identifying information (PII) or any other type of sensitive data.

If the transmission of sensitive or confidential content is required, it is still possible to secure communication between relay agents and servers. Relay agents and servers communicating with relay agents must support the use of IPsec Encapsulating Security Payload (ESP) with encryption in transport mode, according to Section 3.1.1 of [RFC4303] and Section 21.1 of [RFC3315]. Sadly, this requirement is almost universally ignored in real deployments. Even if the communication path between relay agents and server is secured, the path between clients and relay agents or server is not.

Unless underlying transmission technology provides a secure transmission channel, the DHCPv6 options SHOULD NOT include PII or other sensitive information. If there are special circumstances that warrant sending such information over unsecured DHCPv6, the dangers MUST be clearly discussed in security considerations.

25. IANA Considerations

This document has no actions for IANA.

26. Acknowledgements

Authors would like to thank Simon Perreault, Bernie Volz, Ted Lemon, Bud Millwood, Ralph Droms, Barry Leiba, Benoit Claise, Brian Haberman, Richard Barnes, Stephen Farrell and Steward Bryant for their comments.

27. References

27.1. Normative References

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

27.2. Informative References

- [I-D.ietf-dhc-v4configuration]
Rajtar, B. and I. Farrer, "Provisioning IPv4 Configuration Over IPv6 Only Networks", draft-ietf-dhc-v4configuration-03 (work in progress), December 2013.
- [I-D.ietf-softwire-4rd]
Despres, R., Jiang, S., Penno, R., Lee, Y., Chen, G., and M. Chen, "IPv4 Residual Deployment via IPv6 - a Stateless Solution (4rd)", draft-ietf-softwire-4rd-07 (work in progress), October 2013.
- [I-D.ietf-softwire-map-dhcp]
Mrugalski, T., Troan, O., Dec, W., Bao, C., leaf.yeh.sdo@gmail.com, l., and X. Deng, "DHCPv6 Options for configuration of Softwire Address and Port Mapped Clients", draft-ietf-softwire-map-dhcp-06 (work in progress), November 2013.
- [RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.
- [RFC3319] Schulzrinne, H. and B. Volz, "Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers", RFC 3319, July 2003.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.
- [RFC3898] Kalusivalingam, V., "Network Information Service (NIS) Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3898, October 2004.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [RFC4075] Kalusivalingam, V., "Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6", RFC 4075, May 2005.
- [RFC4085] Plonka, D., "Embedding Globally-Routable Internet Addresses Considered Harmful", BCP 105, RFC 4085, June 2005.
- [RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 4242, November 2005.
- [RFC4280] Chowdhury, K., Yegani, P., and L. Madour, "Dynamic Host Configuration Protocol (DHCP) Options for Broadcast and Multicast Control Servers", RFC 4280, November 2005.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [RFC4436] Aboba, B., Carlson, J., and S. Cheshire, "Detecting Network Attachment in IPv4 (DNav4)", RFC 4436, March 2006.
- [RFC4704] Volz, B., "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option", RFC 4704, October 2006.
- [RFC4833] Lear, E. and P. Eggert, "Timezone Options for DHCP", RFC 4833, April 2007.
- [RFC4957] Krishnan, S., Montavont, N., Njedjou, E., Veerepalli, S., and A. Yegin, "Link-Layer Event Notifications for Detecting Network Attachments", RFC 4957, August 2007.

- [RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng, "DHCPv6 Leasequery", RFC 5007, September 2007.
- [RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.
- [RFC5223] Schulzrinne, H., Polk, J., and H. Tschafenig, "Discovering Location-to-Service Translation (LoST) Servers Using the Dynamic Host Configuration Protocol (DHCP)", RFC 5223, August 2008.
- [RFC5460] Stapp, M., "DHCPv6 Bulk Leasequery", RFC 5460, February 2009.
- [RFC5908] Gayraud, R. and B. Lourdelet, "Network Time Protocol (NTP) Server Option for DHCPv6", RFC 5908, June 2010.
- [RFC5970] Huth, T., Freimann, J., Zimmer, V., and D. Thaler, "DHCPv6 Options for Network Boot", RFC 5970, September 2010.
- [RFC5986] Thomson, M. and J. Winterbottom, "Discovering the Local Location Information Server (LIS)", RFC 5986, September 2010.
- [RFC6059] Krishnan, S. and G. Daley, "Simple Procedures for Detecting Network Attachment in IPv6", RFC 6059, November 2010.
- [RFC6334] Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite", RFC 6334, August 2011.
- [RFC6422] Lemon, T. and Q. Wu, "Relay-Supplied DHCP Options", RFC 6422, December 2011.
- [RFC6440] Zorn, G., Wu, Q., and Y. Wang, "The EAP Re-authentication Protocol (ERP) Local Domain Name DHCPv6 Option", RFC 6440, December 2011.
- [RFC6603] Korhonen, J., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", RFC 6603, May 2012.
- [RFC6610] Jang, H., Yegin, A., Chowdhury, K., Choi, J., and T. Lemon, "DHCP Options for Home Information Discovery in Mobile IPv6 (MIPv6)", RFC 6610, May 2012.

[RFC6644] Evans, D., Droms, R., and S. Jiang, "Rebind Capability in DHCPv6 Reconfigure Messages", RFC 6644, July 2012.

[iana] IANA, , "DHCPv6 parameters (IANA webpage)", November 2003, <<http://www.iana.org/assignments/dhcpv6-parameters/>>.

Authors' Addresses

David W. Hankins
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
USA

Email: dhankins@google.com

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

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

Marcin Siodelski
950 Charter Street
Redwood City, CA 94063
USA

Phone: +1 650 423 1431
Email: msiodelski@gmail.com

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

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

Email: suresh.krishnan@ericsson.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 18, 2013

T. Lemon
Nominum
July 17, 2012

Populating the DNS Reverse Tree for DHCP Delegated Prefixes
draft-lemon-dhc-dns-pd-01.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 January 18, 2013.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Terminology	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. Encoding of options	15
9.1. Prefix Delegation Method Types	15
9.2. Prefix Delegation Zone Preference Option	15
9.3. Prefix Delegation Zone Method Option	15
9.4. Prefix Delegation Zone Server Option	15
10. Security Considerations	17
11. IANA Considerations	18
12. References	19
12.1. Normative References	19
12.2. Informative References	19
Author's Address	20

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 1.8×10^{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:

- o 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.
- o An authoritative name server running on the requesting router.
- o 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 authoritative name servers for 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 delegating 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 pprefix. 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.

In all cases where the delegating router updates the delegation for the zone, it must first query the name server or servers listed in the PDZS option for an SOA record for each delegated zone. If the name server does not respond within the standard timeout period, or does not provide an authoritative answer, the delegating router MUST NOT add a delegation for that name server.

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. Encoding of options

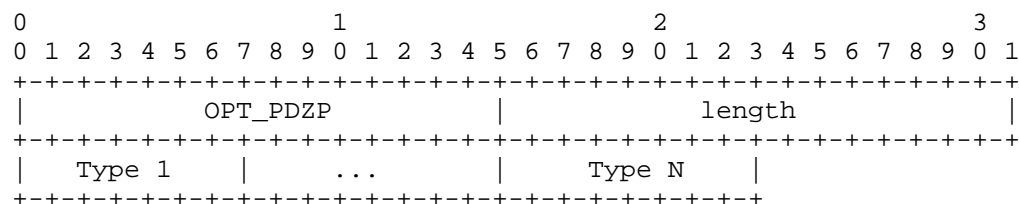
9.1. Prefix Delegation Method Types

Prefix delegation methods are encoded as numbers. Currently three prefix delegation methods are defined:

- 0 Site-Managed
- 1 Provider-managed
- 2 Provider-managed spoofed reverse tree

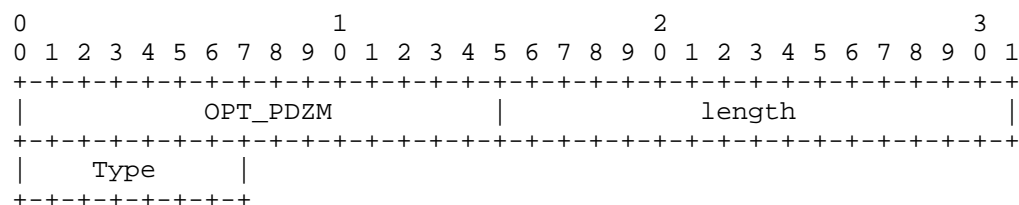
9.2. Prefix Delegation Zone Preference Option

The Prefix Delegation Zone Preference option consists of an option code, OPT_PDZP, followed by a length, followed by one or more Prefix Delegation method type codes.



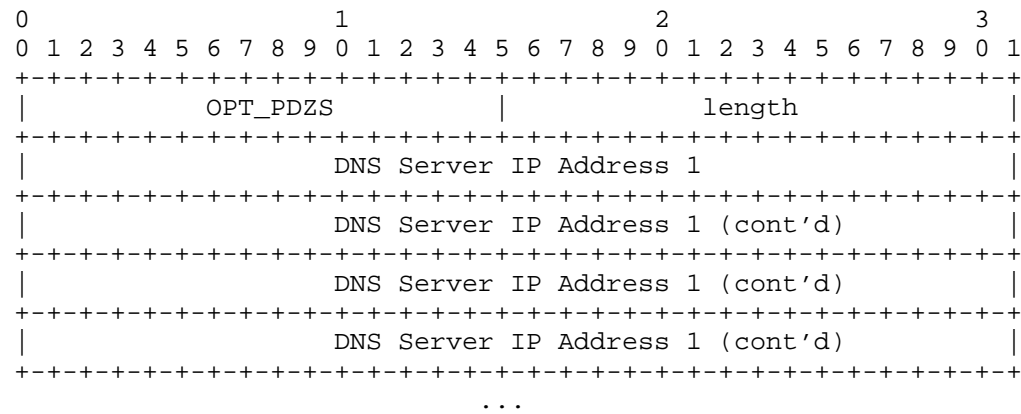
9.3. Prefix Delegation Zone Method Option

The Prefix Delegation Zone Method option consists of an option code, OPT_PDZM, followed by a length, followed by one Prefix Delegation method type code.



9.4. Prefix Delegation Zone Server Option

The Prefix Delegation Zone Server option consists of an option code, OPT_PDZS, followed by a length, followed by zero or more IPv6 addresses.



10. Security Considerations

Some ISPs may have concerns about allowing site-managed DNS subdelegations for the reverse zone, but this concern is a policy issue, not a security issue. In the presence of properly agreed-to terms of service, population of a reverse tree by the end-user is simply a value-added service the ISP may or may not choose to provide. Even in the absence of a legally binding ToS agreement, the worse an end-user could do would be to publish nasty words or bogus PTR records, neither of which is a security concern.

If an implementation were to fail to follow the advice on validating authoritative name servers supplied by the requesting router, it would probably be possible for a coordinated set of requesting routers to perform a DDoS attack on a target by arranging for various entities on the network to query the reverse tree for one or more of the IP addresses in the delegated prefix. However, this would require, first, that the implementation not follow the specification, and second, a fairly complicated setup. In practice, there are easier ways to get a DDoS amplification.

11. IANA Considerations

We request that IANA assign three new option codes from the DHCP Option Codes table of the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) parameters registry maintained in <http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xml>. These option codes will be assigned to the Prefix Delegation Zone Preference (OPT_PDZP), Prefix Delegation Zone Method (OPT_PDZM) and Prefix Delegation Zone Servers (OPT_PDZS) options.

We also request that the IANA add a new table, the Prefix Delegation Zone Method Types table, to the same registry. The first three entries in the table will contain the values specified in the section above titled "Prefix Delegation Zone Method Types." New entries to the table may be added according to the "Specification Required" IANA policy [RFC5226].

12. References

12.1. Normative References

- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

12.2. Informative References

- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

Author's Address

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

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

Dynamic Host Configuration WG
Internet-Draft
Intended status: Standards Track
Expires: March 30, 2013

T. Mrugalski
ISC
P. Wu
Tsinghua University
September 26, 2012

Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for DHCPv4
over IPv6 Endpoint
draft-mrugalski-softwire-dhcpv4-over-v6-option-01

Abstract

[I-D.ietf-dhc-dhcpv4-over-ipv6] defines a way for communication between legacy DHCPv4 clients with DHCPv4 servers over IPv6-only transport. It requires deployment of Client Relay Agent (CRA) that transmits messages to IPv6-Transport Server (TSV) or IPv6-Transport Relay Agent (TRA). Deployed CRA must know an address of TSV or TRA to forward incoming client's messages. This document defines a DHCPv6 option that may be used to provision TSV or TRA location to CRAs.

Status of this Memo

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

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

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

This Internet-Draft will expire on March 30, 2013.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements Language	3
2. Introduction	3
3. The DHCPv4-Over-IPv6 Endpoint DHCPv6 Option	3
4. DHCPv6 Server Behavior	4
5. DHCPv6 Client Behavior	5
6. Security Considerations	5
7. IANA Considerations	6
8. Acknowledgements	6
9. References	6
9.1. Normative References	6
9.2. Informative References	6
Authors' Addresses	7

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. Introduction

[I-D.ietf-dhc-dhcpv4-over-ipv6] defines a way for communication between legacy DHCPv4 clients with DHCPv4 servers (defined in [RFC2131]) over IPv6-only transport. It requires deployment of Client Relay Agent (CRA) that transmits messages to IPv6-Transport Server (TSV) or IPv6-Transport Relay Agent (TRA). Although there are several scenarios envisaged, all of them assume that CRA needs to know the recipient address of the DHCPv4-over-IPv6 traffic. In a typical deployment as [I-D.cui-software-b4-translated-ds-lite] or [I-D.ietf-software-public-4over6], CRA functionality will be a part of a Lightweight B4 element (Basic Bridging BroadBand element) implementation.

Depending on the scenario discussed, the DHCPv4 over IPv6 transport endpoint could be either an IPv6-Transport Server (TSV) or an IPv6-Transport Relay Agent (TRA). Both cases are indistinguishable from CRA perspective. CRA needs to know TSV's or TRA's IPv6 address in advance to relay traffic. Again, the typical envisaged use would be the Lightweight 4over6 architecture, where TSV or TRA could be part of Lightweight AFTR implementation.

As CRA uplink is IPv6-only (otherwise there would be no need to deploy DHCPv4 over IPv6), the only feasible way to provision information to CRA is over DHCPv6. Therefore this document specifies a DHCPv6 option that conveys necessary information.

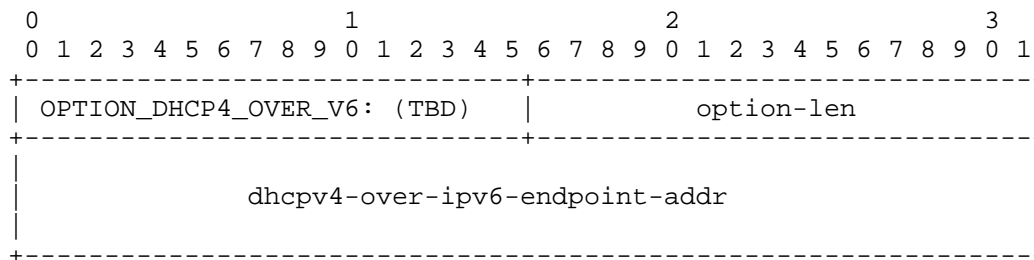
To provide the conveyance of the configuration information, a single DHCPv6 [RFC3315] option is used, expressing the TRA's or TSV's IPv6 address to the CRA.

3. The DHCPv4-Over-IPv6 Endpoint DHCPv6 Option

The DHCPv4-over-IPv6 option is a DHCPv6 option. It consists of an option-code and option-len fields (as all DHCPv6 options have), and a fixed-length dhcpv4-over-ipv6-endpoint-addr field containing an IPv6 address that refers to the DHCPv4 over IPv6 transport endpoint to which the CRA MAY transport DHCPv4 traffic. This address represents TRA or TSV, depending on deployment scenario.

The DHCPv4-over-IPv6 option SHOULD NOT appear in any other than the following DHCPv6 messages: Solicit, Advertise, Request, Renew, Rebind, Information-Request and Reply.

The format of the DHCPv4 over IPv6 option is shown in the following figure:



OPTION_DHCP4_OVER_V6: (TBD)

option-len: 16

dhcpv4-over-ipv6-endpoint-addr: An IPv6 address of the DHCPv4-over-IPv6 transport endpoint.

Figure 1: AFTR-Name DHCPv6 Option Format

The option is validated by confirming that all of the following conditions are met:

1. the option-len is exactly 16;
2. the dhcpv4-over-ipv6-endpoint-addr contains valid unicast address. In particular any address (::), multicast (ff::/8) or IPv4-mapped IPv6 address is invalid and MUST be rejected.

4. DHCPv6 Server Behavior

A DHCPv6 server SHOULD NOT send more than one DHCPv4-over-IPv6 option. It SHOULD NOT permit the configuration of multiple addresses within one DHCPv4-over-IPv6 option. Both of these conditions are handled as errors by the client, so an operator using software that does not perform these validations should be careful not to configure multiple addresses.

RFC 3315 Section 17.2.2 [RFC3315] describes how a DHCPv6 client and server negotiate configuration values using the Option Request Option (OPTION_ORO). As a convenience to the reader, we mention here that a

server will not reply with a DHCPv4-over-IPv6 option if the client has not explicitly enumerated it on its Option Request Option. In other words, server SHOULD send this option only if client explicitly requested it in ORO.

5. DHCPv6 Client Behavior

A client that supports the DHCPv4 over IPv6 functionality of and conforms to this specification MUST include OPTION_DHCP4_OVER_V6 on its OPTION_ORO.

If the client receives the DHCPv4-over-IPv6 option, it MUST verify the option contents as described in Section 3.

If the CRA entity receives more than one DHCPv4-over-IPv6 option, it MUST use only one instance of that option.

If the DHCPv4-over-IPv6 option contains more than one address, the CRA entity system MUST ignore the option. It SHOULD warn its operator about such condition.

Note that a CRA system may have multiple network interfaces, and these interfaces may be configured differently; some may be connected to networks that call for DHCPv4-over-IPv6, and some may be connected to networks that are using normal dual stack or other means. The CRA entity should approach this specification on an interface-by-interface basis. For example, if the CRA entity is attached to multiple networks that provide the DHCPv4-over-IPv6 option, then the CRA entity MUST configure a DHCPv4 over IPv6 transport for each interface separately as each transport provides IPv4 connectivity for each distinct interface.

6. Security Considerations

This document does not present any new security issues, but as with all DHCPv6-derived configuration state, it is completely possible that the configuration is being delivered by a third party (Man In The Middle). As such, there is no basis to trust that the access the DHCPv4-over-IPv6 connection provides can be trusted. It should be protected by available security mechanisms such as IP firewalls.

It should be noted that DHCPv4 over IPv6 traffic may bypass existing firewalls that are typically configured to drop incoming outside DHCPv4 over IPv4 and DHCPv6 over IPv6 traffic.

RFC 3315 [RFC3315] discusses DHCPv6-related security issues.

[I-D.ietf-dhc-dhcpv4-over-ipv6] discusses DHCPv4-over-IPv6 related security issues.

7. IANA Considerations

IANA is kindly requested to allocate DHCPv6 option code TBD to the OPTION_DHCP4_OVER_V6. The value should be added to the DHCPv6 option code space defined in Section 24.3 of [RFC3315].

8. Acknowledgements

Authors would like to thank nobody so far, as we have not received any comments yet.

This work has been partially supported 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).

9. References

9.1. Normative References

- [I-D.ietf-dhc-dhcpv4-over-ipv6]
Cui, Y., Wu, P., Wu, J., and T. Lemon, "DHCPv4 over IPv6 Transport", draft-ietf-dhc-dhcpv4-over-ipv6-05 (work in progress), September 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

9.2. Informative References

- [I-D.cui-software-b4-translated-ds-lite]
Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture", draft-cui-software-b4-translated-ds-lite-08 (work in progress), September 2012.

[I-D.ietf-softwire-public-4over6]

Cui, Y., Wu, J., Wu, P., Vautrin, O., and Y. Lee, "Public
IPv4 over IPv6 Access Network",
draft-ietf-softwire-public-4over6-03 (work in progress),
August 2012.

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

Peng Wu
Tsinghua University
Beijing 100084
P.R.China

Email: pengwu.thu@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 30, 2012

P. Wu
Tsinghua University
Y. Lee
Comcast
Q. Sun
China Telecom
T. Lemon
Nominum, Inc.
April 28, 2012

Dynamic Host Configuration Protocol (DHCP) Options for Port Set
Assignment
draft-wu-dhc-port-set-option-00

Abstract

Due to the exhaustion of global IPv4 address space, there are demands arising for IPv4 address sharing between end users. In such context, different users can employ the same address, but different ports. This document defines two DHCP options for assigning a set of ports to a device. One is used for allocating continuous port set, while the other is designed for non-continuous port set allocation.

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 30, 2012.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	3
2. Requirements Language	4
3. DHCP Option Format	5
3.1. Continuous Port Set Option	5
3.2. Noncontinuous Port Set Option	5
4. Option Examples	8
4.1. Continuous Port Set Option Example	8
4.2. Noncontinuous Port Set Option Example	8
5. Server Behavior	10
6. Client Behavior	11
7. Security Consideration	12
8. IANA Consideration	13
9. References	14
9.1. Normative References	14
9.2. Informative References	14
Authors' Addresses	15

1. Introduction

Due to the exhaustion of global IPv4 address space, there are demands arising for IPv4 address sharing between end users, especially in IPv4-over-IPv6 scenarios. With address sharing, different users can employ the same address, but different port space. In such cases, during the address provisioning process, the port numbers a user device can use should be allocated as well.

This document defines two DHCPv4 options to carry the specific parameters for port set assignment. The Continuous Port Set Option is used for allocating continuous port set, while the Noncontinuous Port Set Option is designed for non-continuous port set allocation.

2. 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 [RFC2119].

3. DHCP Option Format

The format and usage of the two options are defined in the following sections.

3.1. Continuous Port Set Option

This option specifies the min and max port number assigned to a DHCP client, which determines a continuous port range. Figure 1 shows the bit-representation of the option.

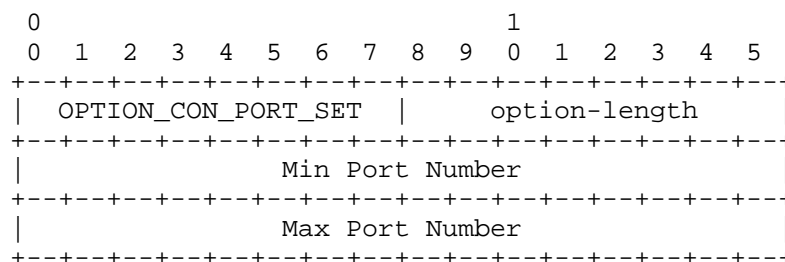


Figure 1 Continuous Port Set Option Format

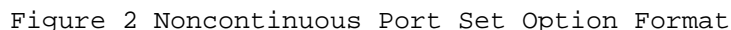
- o option-code: OPTION_CON_PORT_SET(TBD1)
- o option-length: An 8-bit field indicating the length of the option excluding the 'Option Code' and the 'Option Length' fields. In this option, its value is 4 octets.
- o Min Port Number: The minimum port number in the port range. The value of Min Port Number MUST be within 0~65535.
- o Max Port Number: The maximum port number in the port range. The value of Max Port Number MUST be within 0~65535, and not smaller than the value of Min Port Number.

Section 4.1 further explains the above parameters with an example.

3.2. Noncontinuous Port Set Option

There can be requests for noncontinuous port set. This option caters to such requirements. In this option, the PSID is short for Port-Set ID which identifies a set of ports exclusively assigned to a device. It is defined in the MAP draft

[I-D.mdt-software-mapping-address-and-port], and so are PSID Offset and the parameters of (a,k,m) used below. Figure 2 shows the format of the Noncontinuous Port Set Option.



- In the context of noncontinuous port set, as is defined in Section 5.1.1 of [I-D.mdt-softwire-mapping-address-and-port], the port number consist of Port Range Index ($A(j)$ in Figure 3, a bits), PSID (k bits) and Continuous Port Index ($M(i)$ in Figure 3, m bits). For the readers' convenience, the format of the port number is included in this draft as well. $i, j, A(j)$ and $M(i)$ are the same as the definition in the GMA port mapping algorithm [I-D.mdt-softwire-mapping-address-and-port]

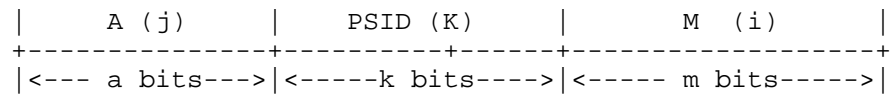


Figure 3 Bit Representation of a GMA port number

If PSID Offset is greater than 0 ($a > 0$), j MUST be larger than 0, in order to exclude the system ports ([I-D.ietf-tsvwg-iana-ports]) or ports saved by SPs. If $a = 0$, j may be 0 to allow the provisioning of the reserved ports. With a given sharing ratio (2^k) and the PSID value, the ports assigned to a client can be calculated by increasing i and j continuously. Section 4.2 explains the algorithm further with an example.

4. Option Examples

4.1. Continuous Port Set Option Example

A Continuous Port Set Option example with the assigned port range 4096~8191 is as follows. There is no specific requirement on the port number format.

```

      0                               1
      0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
+---+---+---+---+---+---+---+---+---+---+---+---+
| OPTION_CON_PORT_SET |                               |
+---+---+---+---+---+---+---+---+---+---+---+---+
|                               4096                    |
+---+---+---+---+---+---+---+---+---+---+---+---+
|                               8191                    |
+---+---+---+---+---+---+---+---+---+---+---+---+

```

Example 1 Continuous Port Set Option Example

4.2. Noncontinuous Port Set Option Example

Here is an example of Noncontinuous Port Set Option, with PSID offset 4, PSID length 10 and PSID value 1021 (i.e. a = 4, k = 10 and PSID = 1021):

```

      0                               1
      0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
+---+---+---+---+---+---+---+---+---+---+---+---+
| OPTION_NCON_PORT_SET |                               |
+---+---+---+---+---+---+---+---+---+---+---+---+
|           4           |           10           |
+---+---+---+---+---+---+---+---+---+---+---+---+
| 1  1  1  1  1  1  1  1  0  1  0  0  0  0  0  0 |
+---+---+---+---+---+---+---+---+---+---+---+---+

```

Example 2 Noncontinuous Port Set Option Example (a = 4, k = 10, PSID = 1021)

The first 10 bits of the last two octets(11 1111 1101) are the value of PSID. And the allocated port ranges are:

```

      Port-range-1                               Port-range-2
PSID=1021| 8180, 8181, 8182, 8183, | 12276, 12277, 12278, 12279,| ...

```

All these port ranges form the full port set.

The port set calculation procedure of a client when receiving the parameters of (a,k,PSID) follows the GMA algorithm proposed in section 5.1 of [I-D.mdt-softwire-mapping-address-and-port]. Two examples in [I-D.mdt-softwire-mapping-address-and-port] are illustrated here for the readers' convenience.

For sharing ratio 1024, PSID offset a = 4 and PSID length k = 10

	Port-range-1	Port-range-2
PSID=0	4096, 4097, 4098, 4099,	8192, 8193, 8194, 8195, ...
PSID=1	4100, 4101, 4102, 4103,	8196, 8197, 8198, 8199, ...
PSID=2	4104, 4105, 4106, 4107,	8200, 8201, 8202, 8203, ...
PSID=3	4108, 4109, 4110, 4111,	8204, 8205, 8206, 8207, ...
...		
PSID=1023	8188, 8189, 8190, 8191,	12284, 12285, 12286, 12287, ...

Example 3: GMA calculation with a = 4, k = 10

For sharing ratio 64, PSID offset a = 0 and PSID length k = 6

	Port-set
PSID=0	[0 - 1023]
PSID=1	[1024 - 2047]
PSID=2	[2048 - 3071]
PSID=3	[3072 - 4095]
...	
PSID=63	[64512 - 65535]

Example 4: GMA calculation with a = 0, k = 6

5. Server Behavior

The server will not reply with either of the two options until the client has explicitly listed one of them in the Parameter Request List (Option 55).

Server MUST reply with Continuous Port Set Option if the client requested `OPTION_CON_PORT_SET` in its Parameter Request List. Server MUST reply with Noncontinuous Port Set Option if the client requested `OPTION_NCON_PORT_SET` in its Parameter Request List. The server MUST run an address & port-set pool which plays the same role as address pool in regular DHCP server. If the server supports Noncontinuous Port Set Option, address & port-set pool MUST follow the GMA-format port-set.

The port-set assignment SHOULD be coupled with the address assignment process. Therefore server SHOULD assign the address and port set in the same DHCP messages. and the lease information for the address is applicable to the port-set as well.

6. Client Behavior

The DHCP client applying for the a port-set MUST include either the `OPTION_CON_PORT_SET` or `OPTION_NCON_PORT_SET` code in the Parameter Request List (Option 55). If the client requests the `OPTION_CON_PORT_SET`, it will retrieve a Continuous Port Set Option and use the ports ranging from Min port number to Max port number. If the client requests `OPTION_NCON_PORT_SET` and retrieves a Noncontinuous Port Set Option, its port set follows the specific port number format defined in section 5.1.1 of MAP draft [I-D.mdt-software-mapping-address-and-port]. The client derives the PSID offset (a bits), PSID length (k bits) and the PSID from the option, and performs GMA to get the precise port set. The client renews or releases the DHCP lease with the port set.

7. Security Consideration

This specification raises no particular security issues to the DHCPv4 protocol model.

8. IANA Consideration

IANA is kindly requested to allocate DHCP option codes to the `OPTION_CON_PORT_SET` and `OPTION_NCON_PORT_SET`. Both codes should be added to the DHCP option code space.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.
- [RFC3527] Kinnear, K., Stapp, M., Johnson, R., and J. Kumarasamy, "Link Selection sub-option for the Relay Agent Information Option for DHCPv4", RFC 3527, April 2003.
- [RFC4925] Li, X., Dawkins, S., Ward, D., and A. Durand, "Softwire Problem Statement", RFC 4925, July 2007.

9.2. Informative References

- [I-D.ietf-tsvwg-iana-ports]
Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", draft-ietf-tsvwg-iana-ports-10 (work in progress), February 2011.
- [I-D.mdt-softwire-mapping-address-and-port]
Bao, C., Troan, O., Matsushima, S., Murakami, T., and X. Li, "Mapping of Address and Port (MAP)", draft-mdt-softwire-mapping-address-and-port-03 (work in progress), January 2012.

Authors' Addresses

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

Phone: +86-10-6278-5822
Email: peng-wu@foxmail.com

Yiu L. Lee
Comcast
One Comcast Center
Philadelphia PA 19103
USA

Phone:
Email: yiu_lee@cable.comcast.com

Qiong Sun
China Telecom
Room 708, No.118, Xizhimennei Street
Beijing 100035
P.R.China

Phone: +86-10-58552936
Email: sunqiong@ctbri.com.cn

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

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

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 13, 2013

T. Yang
L. Li
Q. Ma
China Mobile
July 12, 2012

Mitigating aggregated traffic of DHCP discover messages
draft-yang-dhc-ipv4-dis-01

Abstract

This document defines a new option DIS_MAX_RT which can mitigate aggregated traffic caused by discover messages the clients send to the server.

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

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

This Internet-Draft will expire on January 13, 2013.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Requirements Language	3
3. Potential Problems	3
4. DIS_MAX_RT and DIS_MAX_RT_OPTION	6
5. Security Considerations	7
6. IANA Considerations	7
7. References	7
8. Acknowledgement	7
Authors' Addresses	7

1. Introduction

In RFC2131 [RFC2131], there are no specific definitions for client's operation if the server does not respond for the discover messages. In some cases, this will lead to an unacceptably high volum of aggregated traffic at a DHCPv4 server.

In RFC3315 [RFC3315], SOL_MAX_RT is defined as an option of DHCPv6 message to prevent the frequently requesting of clients, which reduce the aggregated traffic. In DHCPv4, there are no corresponding IPv4 options. Although the format of DHCPv4 is different with DHCPv6, it is also necessary to introduce similar option in DHCPv4 to keep the consistency between DHCPv4 and DHCPv6.

This document updates RFC 2131 [RFC2131] by defining a new option DIS_MAX_RT which makes the DHCPv4 server mitigating aggregated traffic of client's discover messages.

2. 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 [RFC2119].

3. Potential Problems

RFC2131 [RFC2131] defines the interaction between the DHCP server and clients. There are no specific discription for client's operation when the client does not receive the DHCPOFFER in response to its DHCPDISCOVER message. In normal IPv4 environment, clients will flood DHCPDISCOVER messages only when the server or link is broken. But in Dual-Stack scenarios, the problem becomes more frequent and serious. In IPv6 LAN/WLAN network or intranet, the core router or AC often plays the role of DHCP server, and the clients are serval thousands PC or mobile phones. If the server is configured in IPv6-only, the clients in dual-stack or IPv4-only, they will broadcast DHCPDISCOVER messages endlessly in the LAN or WLAN. The thousands clients will cause a DDOS-like attack to all the servers in the intranet.

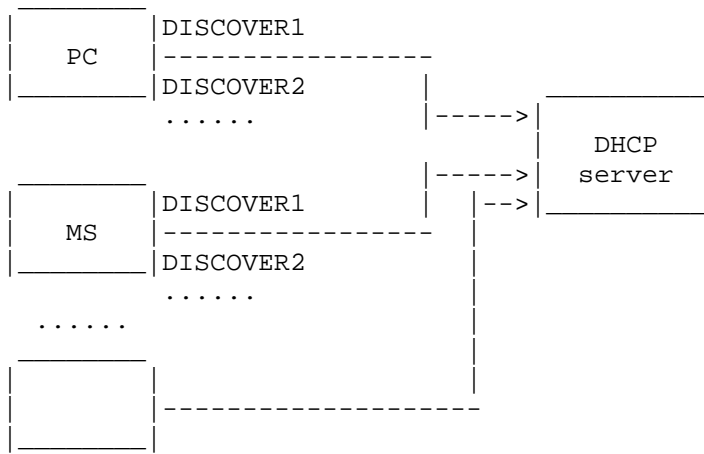


Figure 1: DHCPDISCOVER flood in LAN/WLAN

To avoid this problem, most of the terminals creat backoff algorithms which can help them retransmit DHCPDISCOVER message in different frequency according to their state machine in different Operating Systems, because there is no specific defenition in RFCs to restrict the terminals behaviors when the server is down or in a dual-stack scenario as discripted upwards. But the same point of almost all the verious Operating Systems is that they could not stop DHCPDISCOVER requests enven to an IPv6-only server. We test some of the most popular terminals' OS in WLAN, the results are illuminated as below.

DHCP Discovery Packages Time Table										
No	Windows7		Windows XP		IOS_5.0.1		Android_2.3.7		Symbian_S60	
	Time	Time offset	Time	Time offset	Time	Time offset	Time	Time offset	Time	Time offset
1	0		0		0.1		7.8		0	
2	3.9	3.9	0.1	0.1	1.4	1.3	10.3	2.5	2	2
3	13.3	9.4	4.1	4	3.8	2.4	17.9	7.6	6	4
4	30.5	17.2	12.1	8	7.9	4.1	33.9	16	8	2
5	62.8	32.3	29.1	17	16.3	8.4	36.5	2.6	12	4
6	65.9	3.1	64.9	35.8	24.9	8.6	reconnect		14	2
7	74.9	9	68.9	4	33.4	8.5	56.6	20.1	18	4
8	92.1	17.2	77.9	9	42.2	8.8	60.2	3.6	20	2
9	395.2	303.1	93.9	16	50.8	8.6	68.4	8.2	24	4
10	399.1	3.9	433.9	340	59.1	8.3	84.8	16.4	26	2
11	407.1	8	438.9	5	127.3	68.2	86.7	1.9	30.1	4.1
12	423.4	16.3	447.9	9	128.9	1.6	reconnect		32.1	2
13	455.4	32	464.9	17	131.1	2.2	106.7	20	36.1	4
14	460.4	5	794.9	330	135.1	4	111.4	4.7	38.1	2
15	467.4	7	799.9	5	143.4	8.3	120.6	9.2	42.1	4
16	483.4	16	808.9	9	151.7	8.3	134.9	14.3	44.1	2
17	842.9	359.5	824.9	16	160.4	8.7	136.8	1.9	48.2	4.1
18	846.9	4	1141.9	317	168.8	8.4	reconnect		50.2	2

Terminals DHCPDISCOVER requests when
Server's DHCP module is down

figure 2. Terminals DHCPDISCOVER requests when Server's DHCP module is down For Windows7, it seems to initiate 8 times DHCPDISCOVER requests in about 300s interval. For WindowsXP, firstly it launches 9 times DHCPDISCOVER messages, but after that it cannot get any response from the server, then it initiates 5 times requests in one cycle in around 330s intervals, and never stop. For IOS5.0.1, it seems like WindowsXP. There are 10 times attempts in one cycle, and the interval is about 68s. Symbian_S60 uses the simplest backoff method, it launches DISCOVER in every 2 or 4 seconds. Android2.3.7 is the only Operating System which can stop DISCOVER request by disconnect its wireless connection. It reboot wireless and dhcp connection every 20 seconds. Obviously, DHCP server needs to weaken the traffic which is like DDoS attack caused by the clients when many DHCPv4 clients send discovery messages incessantly when the DHCPv4 server is configured no respond to discover messages or the IPv4 address pool is empty.

4. DIS_MAX_RT and DIS_MAX_RT_OPTION

In our experiments described upwards, some of the most popular OS will send several discover messages every 1 or 5 minutes, and send the message endlessly. So the DHCP server needs a mechanism to weaken the traffic.

It is necessary to define an uniform identification named DIS_MAX_RT for client to follow when it needs to retransmit DHCPDISCOVER. Client should retransmits the message in a period refer to the DIS_MAX_RT value. This parameter can be initiated by client and configured by DHCP server. Client must support this new option, and should deploy some backoff algorithm to avoid launch DISCOVER more frequently. Server must also support this option, and could refill the parameter according to its state.

According to the definition of DHCP option in RFC2132 [RFC2132], a new option named DIS_MAX_RT_OPTION is defined. The format of DIS_MAX_RT_OPTION is:

Code	Len	T1	T2	T3	T4
Code	DIS_MAX_RT_OPTION (TBD).				
Len	4.				
T1-T4	4 octets, Overriding value for DIS_MAX_RT in seconds.				

DIS_MAX_RT_OPTION

The DIS_MAX_RT_OPTION option needs IANA to assign a new Code to indicate and its length (Len value) is 4 octets. From T1 to T4, there are 4 octets space to indicate the max retransmission time period. MRT(T1-T4) identifies the interval time client sends two concatenated DISCOVER message. MRT must > 0; When MRT=FFFF, client should not send DISCOVER any more. A DHCPv4 client MUST include the DIS_MAX_RT_OPTION in any message it sends. The DHCPv4 server MAY include the DIS_MAX_RT_OPTION code in any response it sends to a client that has included the DIS_MAX_RT option code in a request message. The process of this option is described below: 1. Client must initial the time parameter by any random algorithm or any others, and set T1-T4 in DIS_MAX_RT_OPTION. IF client receives DIS_MAX_RT_OPTION from server, it should retransmit DISCOVER according the MRT in the option. As a result of receiving this

option, the DHCPv4 client MUST NOT send any request messages more frequently than allowed by the retransmission mechanism defined by their OS. Client should deploy backoff algorithm to retransmit the message if it does not receive any message from server until the backoff time is triggered. 2. When server receives a request including a DIS_MAX_RT_OPTION, it MAY ignore the value of DIS_MAX_RT and assign a new value in the response to make the client refresh its DIS_MAX_RT. It can change MRT longer than the initialized time if the IPv4 address pool is empty or according to the administrator's configuration. Server can also change the value to FFFF if it does not want to support any more IPv4 address request or in a normal address allocation process in DHCPOFFER or any other messages.

5. Security Considerations

The security problem is under discussion.

6. IANA Considerations

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

7. References

- (1) RFC[2131] Dynamic Host Configuration Protocol
- (2) RFC[2132] DHCP Options and BOOTP Vendor Extensions
- (3) RFC[3315] Dynamic Host Configuration Protocol for IPv6(DHCPv6)
- (4) "draft-droms-dhc-dhcpv6-solmaxrt-update-02" Modification to Default Value of SOL_MAX_RT

8. Acknowledgement

Thanks for the authors of "draft-droms-dhc-dhcpv6-solmaxrt-update-02" and the contributions from Lianyuan Li.

Authors' Addresses

Tianle Yang
China Mobile
32, Xuanwumenxi Ave.
Xicheng District, Beijing 01719
China

Email: yangtianle@chinamobile.com

Li Lianyan
China Mobile
32, Xuanwumenxi Ave.
Xicheng District, Beijing 01719
China

Email: lilianyan@chinamobile.com

Qiongfang Ma
China Mobile
32, Xuanwumenxi Ave.
Xicheng District, Beijing 01719
China

Email: maqiongfang@chinamobile.com

DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 29, 2013

L. Yeh, Ed.
Huawei Technologies
M. Boucadair
France Telecom
T. Lemon
Nominum, Inc
J. Hu
China Telecom
July 28, 2012

Prefix Pool Option for DHCPv6 Relay Agents on Provider Edge Routers
draft-yeh-dhc-dhcpv6-prefix-pool-opt-08

Abstract

The DHCPv6 Prefix Pool option provides a mechanism for DHCPv6 Prefix Delegation (DHCPv6-PD), allowing the DHCPv6 server to notify a DHCPv6 relay agent implemented on a Provider Edge (PE) router about active prefix pools allocated by the DHCPv6 server to the PE router. The information of active prefix pools can be used to enforce IPv6 route aggregation on the PE router by adding or removing aggregated routes according to the status of the prefix pools. The advertising of the aggregated routes in the routing protocol enabled on the network-facing interface of PE routers will dramatically decrease the number of the routing table entries in the ISP 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 January 29, 2013.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Terminology and Conventions	4
3. Scenario and Network Architecture	5
4. Prefix Pool Option	6
5. Relay Agent Behavior	8
6. Server Behavior	9
7. Security Considerations	11
8. IANA Considerations	11
9. Contributors List	11
10. Acknowledgements	11
11. References	12
11.1. Normative References	12
11.2. Informative References	12
Authors' Addresses	12

1. Introduction

The DHCPv6 protocol [RFC3315] specifies a mechanism for the assignment of IPv6 address and configuration information to IPv6 nodes. The DHCPv6 Prefix Delegation (DHCPv6-PD) [RFC3633] specifies a mechanism for the delegation of IPv6 prefixes from the Delegating Router (DR) acting as the DHCPv6 server to the Requesting Routers (RR) acting as the DHCPv6 Clients. DHCPv6 servers always maintain authoritative information associated to their operations including, but not limited to, the binding data of the delegated IPv6 prefixes, the lease data of the delegated IPv6 prefixes, and the status of their prefix pools. A prefix pool configured and maintained on the server can usually be a short prefix (e.g., a /40 prefix) out of which the longer prefixes (e.g., /56 prefixes) are delegated to customer networks.

In the scenario of a centralized DHCPv6 server, the Provider Edge (PE) routers act as DHCPv6 relay agents when the DHCPv6 server and the Customer Edge (CE) router (a.k.a. Routed-RG or Routed-CPE) acting as RRs and DHCPv6 clients are not on the same link. For ensuring reachability, the PE routers always need to add or withdraw the route entries directing to each customer network in their routing table to reflect the status of IPv6 prefixes delegated by the DHCPv6 server to CE routers (see Section 6.2, [BBF TR-177]).

When a routing protocol is enabled on the network-facing interface of the PE router, all the routes directing to the customer networks are advertised in the ISP network. This will make the number of route entries in the routing table on the ISP router be unacceptable large. Hence, it is desirable to aggregate the routes directing to the customer networks on the PE router.

Because the prefixes of the customer networks can not be guaranteed to be active and continuous, the routing protocol enabled on the PE router in general can not create one aggregated route automatically to cover all the prefixes delegated within the prefix pool. One way to make the aggregated routes (e.g., black-hole routes) pointing to each of the prefix pools is to configure them manually and permanently, but the PE router is not really aware about the status of the prefix pools, especially when it acts as the relay agent.

This document proposes a new Prefix Pool option for the DHCPv6 relay agent implemented on PE routers, allowing the DHCPv6 server to notify the DHCPv6 relay agent about the prefix of pools. After the PE router received information about the prefix pools, the aggregated route entries per the provision status of the prefix pools can be added or withdrawn in the routing table of the PE router. The aggregated routes will then be advertised into the ISP network

through the routing protocol enabled on the PE's network-facing interface.

DHCPv6 Bulk Leasequery [RFC5460] specifies a mechanism for bulk transfer of the binding data of each delegated prefix from the server to the requestor (i.e., a DHCPv6 relay agent), to support the replacement or reboot event of a relay agent. In this document, the capability of DHCPv6 Bulk Leasequery will be extended to support the bulk transfer of the status of the prefix pools for route aggregation.

The automatic mechanisms described in this document depends on the existing DHCPv6 protocols and implementations without requiring a new DHCPv6 message or a new interface for the configuration of the aggregated route. The administrator of the ISP network can decide whether to inject the aggregated route or not based on the policies defined on the DHCPv6 server.

2. Terminology and Conventions

This document defines a new DHCPv6 option to communicate the prefix of an IPv6 prefix pool. This document should be read in conjunction with the DHCPv6 specifications, [RFC3315], [RFC3633], [RFC5007] and [RFC5460], for understanding the complete mechanism. Definitions for terms and acronyms not specified in this document are defined in [RFC3315], [RFC3633], [RFC3769], [RFC5007] and [RFC5460].

The following terms can be found in this document:

- o Requesting Router (RR): A router defined in [RFC3633] that acts as a DHCPv6 client, and is requesting prefix(es) to be assigned.
- o Delegating Router (DR): A router defined in [RFC3633] that acts as a DHCPv6 server, and is responding to the prefix request.
- o Prefix Pool: An IPv6 address space allocated with a common prefix out of which the longer prefixes are delegated via prefix delegation.
- o Aggregated Route: A route entry created on an edge router, is based on the knowledge of a delegated prefix pool.
- o Requestor: A router defined in [RFC5007] that acts as a DHCPv6 relay agent, is leasequery client.

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. Scenario and Network Architecture

Figure 1 and Figure 2 illustrate two typical cases of the targeted network architectures.

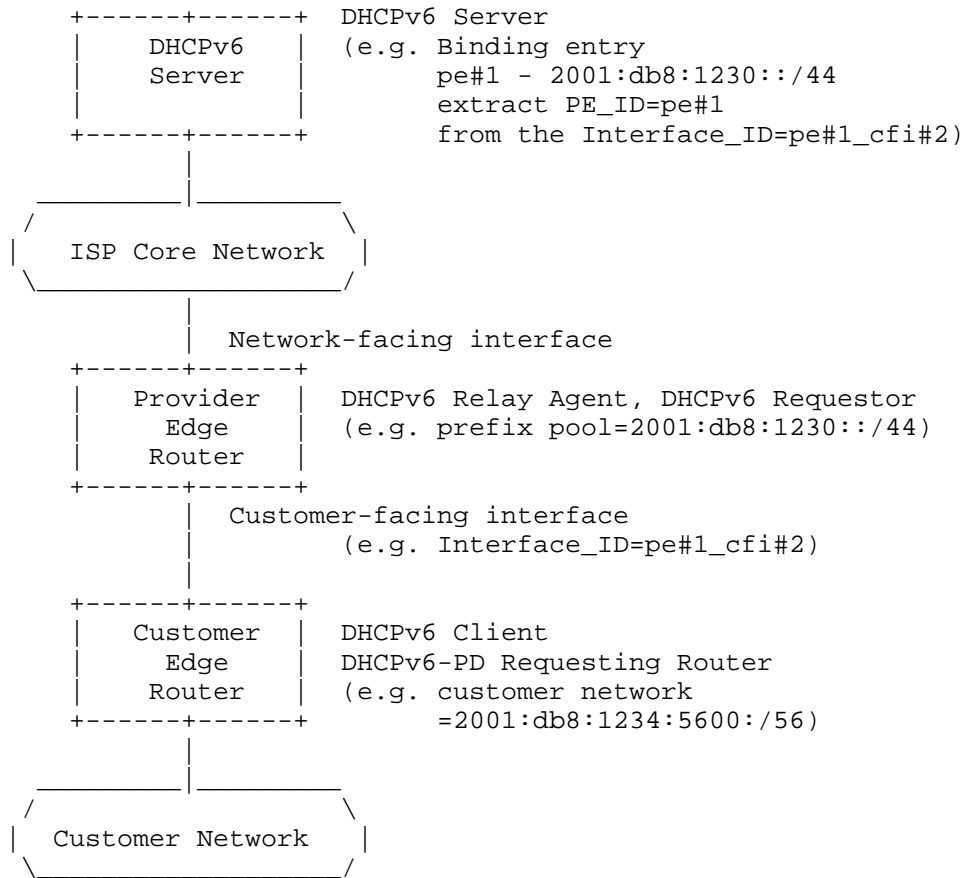


Figure 1: Use case of ISP-Customer network where CPE is directly connected to PE

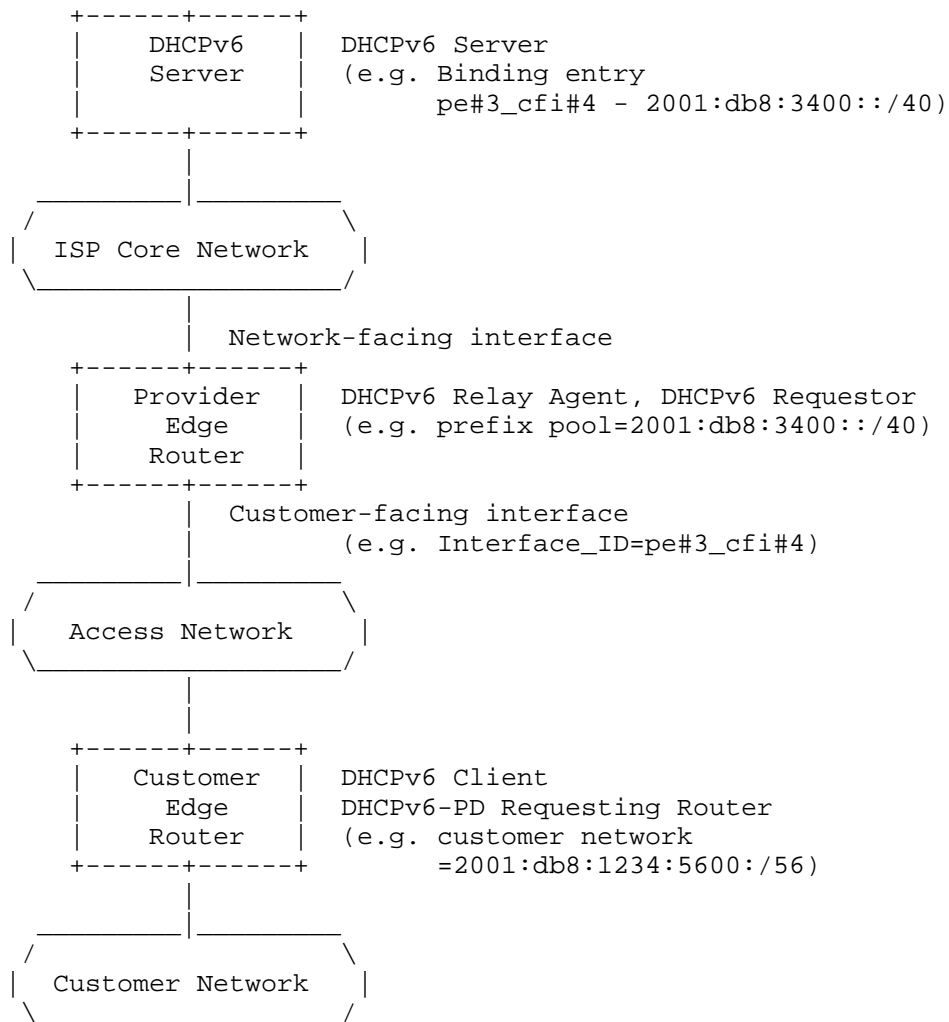
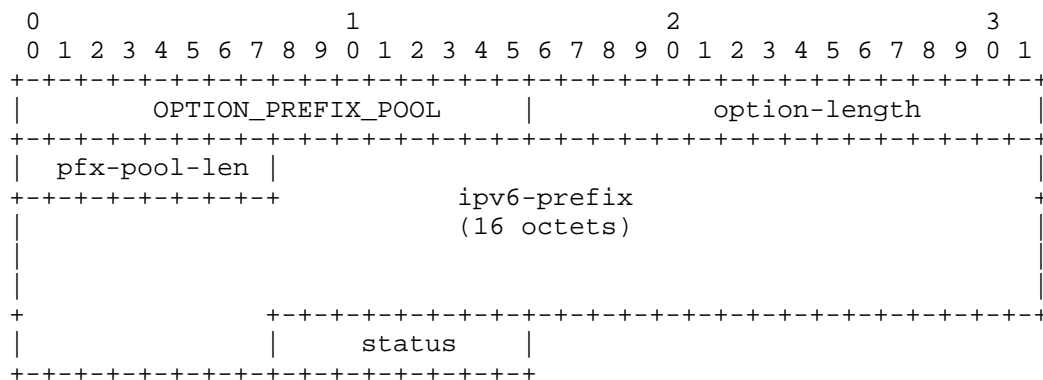


Figure 2: Use case of ISP-Customer network where CPE is connected to PE through access network

4. Prefix Pool Option

The format of the Prefix Pool option is shown in Figure 3.



option-code: OPTION_PREFIX_POOL (TBD)
 option-length: 18
 pfx-pool-len: Length for the prefix pool in bits
 ipv6-prefix: IPv6 prefix of the prefix pool
 status: Status of the prefix pool, indicating the
 availability of the prefix pool maintained
 on the server.

The codes of the status are defined in the following table.

Name	Code
Active	0
Released	1
Reserved	2~255

The 'Active' status of the prefix pool indicated in this option can be used to add the prefix pool and its associated aggregated route on the relay agent; while the 'Released' status of prefix pool indicated in this option can be used to withdraw the prefix pool and its associated aggregated route on the relay agent.

If the administrative policy on the DHCPv6 server permits to support route aggregation on the relay agent, the status of prefix pool can be determined by the delegated prefixes within the associated prefix pool. If there is one delegated prefix within the pool that has a valid lease, the status of the prefix pool will be 'Active'. Otherwise, the status of the prefix pool is 'Released'. If the administrative policy on the server does not permit to support route aggregation on the DHCPv6 relay agent, the status of the prefix pool will always be 'Released'.

Discussion:

The alternative option might include the lease information in the prefix pool, then populate it to relay agent, make the state machine on the relay agent keep synchronizing the lease and status of the associated prefix pool with the server. But the solution proposed in this draft is to let relay agent confirm the received status of the prefix pool by itself as per the leases of delegated customer prefixes within it, and build its own lease for the prefix pool.

5. Relay Agent Behavior

The relay agent who needs the information of prefix pools, must include the associated requested-option-code in Option Request option (OPTION_ORO, 6) to request the Prefix Pool option (OPTION_PREFIX_POOL, [TBD]) from the DHCPv6 server, who maintains the status of the prefix pools associated to the relay agent itself (Figure 1) or its particular customer-facing interface (Figure 2), when receiving the DHCPv6-PD message from clients. The DHCPv6 relay agent can include this Option Request option for the Prefix Pool option in the relay-forward (12) message of SOLICIT (1), REQUEST (3), RENEW(5), REBIND (6) and RELEASE (8). The relay agent may also include the Prefix Pool option with the values of pfx-pool-len and IPv6-prefix to indicate its preference, which the prefix pool the relay agent would like the server to return.

The relay agent should include the Interface ID option (OPTION_INTERFACE_ID, 18) so that the DHCPv6 server can identify the relay agent itself or its particular customer-facing interface to which the prefix pool is associated, if the server would not like to use the link-address field specified in the encapsulation of the DHCPv6 relay-forward message to identify the interface of the link on which the clients are located.

The relay agent may set up a table for the leases or status of the prefix pools on it as per the delegated customer prefixes within it. The lease of the prefix pools must dynamically set to be the maximum lease of the delegated customer prefixes. If there is no route entry directing to the customer network within the aggregated route associated with the prefix pool, the relay agent shall automatically withdraw the aggregated route.

After receiving the Prefix Pool option for the relay agent itself or its particular customer-facing interface in the relay-reply message (13) of REPLY (7) from the DHCPv6 server, the relay agent acting as the PE router shall confirm the status of the prefix pool as per the leases of delegated customer prefixes within it, then add or withdraw the aggregated route entry per the status of the prefix pool. If the

status of the prefix pool received and confirmed is 'Active', the relay agent shall add an aggregated route entry in its routing table, if the same entry has not been added in before. If the status of the prefix pool received is 'Released', the relay agent shall withdraw the associated aggregated route entry in its routing table, if the same entry has not been withdrawn before.

The relay agent advertises its routing table including the entries of the aggregated routes based on the information of prefix pools when the routing protocol is enabled on its network-facing interface.

The Relay Agent (i.e., Requestor) can use the DHCPv6 Bulk Leasequery [RFC5460] to query the binding data of prefix pools in the 'Active' status from the server. After established a TCP connection with the DHCPv6 server, the relay agent must include Query option (OPTION_LQ_QUERY, 44) and set the proper query-type (QUERY_BY_RELAY_ID, QUERY_BY_LINK_ADDRESS, QUERY_BY_REMOTE_ID), link-address and query-options in the LEASEQUERY (14) message. The query options must include Option Request option (OPTION_ORO, 6) to request the Prefix Pool option (OPTION_PREFIX_POOL, [TBD]) from the server.

6. Server Behavior

Per DHCPv6-PD [RFC3633], if the prefix of the customer network requested in relay-forward (12) message of SOLICIT, REQUEST, RENEW, REBIND from the DHCPv6 client (i.e., the RR) has a valid lease, the DHCPv6 server (i.e., the DR) will delegate the prefix with the relevant parameters in the relay-reply (13) message of REPLY. In order to give a meaningful reply, the server has to be able to maintain the binding data of the delegated IPv6 prefixes with the identification of the client. The Interface ID option (OPTION_INTERFACE_ID, 18) nested in the relay-forward message is usually used to identify the access line of the client.

After receiving the Option Request option (OPTION_ORO, 6) requesting the Prefix Pool option (OPTION_PREFIX_POOL, [TBD]) in the relay-forward messages of the PD, the server must include the Prefix Pool option with the status indicated for the associated relay agent itself (Figure 1) or its customer-facing interface (Figure 2) in the relay-reply messages if the relay-forward messages received are valid.

The server may use the link-address specified in relay-forward message to identify the relay agent itself or its particular customer-facing interface where the prefix pool is associated, but the server has to maintain the binding data of prefix pools in association with these link-addresses. To be more readable, the

server can alternatively use the Interface ID option (OPTION_INTERFACE_ID, 18) included in the relay-forward message by the relay agent to identify the relay agent itself (Figure 1) or its particular customer-facing interface (Figure 2) where the prefix pool is associated. In order to give a meaningful reply, the server has to maintain the binding data of prefix pools in association with the information derived from the Interface ID option.

Per DHCPv6 [RFC3315], the server shall copy the same Interface ID option received via the relay-forward message into the relay-reply message.

If the administrative policy on the DHCPv6 server permits to support route aggregation on the relay agent for some particular prefix, the status of prefix pool can be determined by the delegated prefixes within the associated prefix pool. If there is at least one delegated prefix within the pool that has a valid lease, the server shall set the status of the associated prefix pool to be 'Active'. After the last prefix releasing in the associated prefix pool, the server shall set the status of the associated prefix pool to be 'Released'. If the administrative policy on the server does not permit to support route aggregation on the DHCPv6 relay agent, the server shall set the status of the prefix pools always to be 'Released'.

When the administrator of the server changes the setting to support route aggregation on the relay agent for the particular prefix pool, the status of the prefix pool may change from 'Released' to be 'Active' if at least one delegated prefix within the prefix pool has the valid lease. When the administrator of the server changes the setting not to support route aggregation on the relay agent for the particular prefix pool, the status of the prefix pool may change from 'Active' to be 'Released' if at least one delegated prefix within the prefix pool has the valid lease. Then the server may send a relay-reply message of RECONFIGURE (10) to initiate immediately a Renew (5) / Reply (7) PD message exchange with Prefix Pool option between one active client and the server.

Multiple prefix pools may be associated with the same PE router implementing a DHCPv6 relay agent (Figure 1) or its customer-facing interface (Figure 2) in the binding table on the server. Note that the delegated prefix is only from one prefix pool.

After receiving the LEASEQUERY (14) message from the relay agent with the Query option (OPTION_LQ_QUERY, 44) including the Option Request option (OPTION_ORO, 6) to request the Prefix Pool option (OPTION_PREFIX_POOL, [TBD]), the server must include the Client Data options (OPTION_CLIENT_DATA, 45) in the LEASEQUERY-REPLY (15) and

LEASEQUERY-DATA (16) message to convey the binding data of the associated prefix pools with the 'Active' status through the established TCP connection per [RFC5460]. Each Client Data option shall contain a Prefix Pool option, and may contain the Interface ID option (OPTION_INTERFACE_ID, 18). In order to be able to provide meaningful replies to different query types, the server has to be able to maintain the relevant association of prefix pools with the RELAY_ID, link addresses or Remote_ID of the relay agent in its binding database.

7. Security Considerations

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

8. IANA Considerations

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

9. Contributors List

Juergen Schoenwaelder
Jacobs University Bremen
Bremen
Germany

Email: j.schoenwaelder@jacobs-university.de

Tina Tsou
Huawei Technologies
Santa Clara, CA
USA

Email: tina.tsou.zouting@huawei.com

10. Acknowledgements

Thanks to Ralph Droms for the inspiration from his expired draft (RAAN option), to the DHC working group members, Bernie Volz, Ole Troan and Roberta Maglione for the discussion in the mailing list, to

Christian Jacquenet for pointing out the draft should cover one more use case of ISP-Customer network where CPE is directly connected to PE, to Sven Ooghe for some revisions in the email review, to Shrinivas Ashok Joshi for pointing out the draft should cover the robust mechanism against the case of reboot, to Adrian Farrel for the orientation guide on this draft in IETF80 at Prague.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng, "DHCPv6 Leasequery", RFC 5007, September 2007.
- [RFC5460] Stapp, M., "DHCPv6 Bulk Leasequery", RFC 5460, February 2009.

11.2. Informative References

- [BBF TR-177] Broadband Forum, "IPv6 in the context of TR-101, Issue 1", November 2010.
- [RFC3769] Miyakawa, S. and R. Droms, "Requirements for IPv6 Prefix Delegation", RFC 3769, June 2004.

Authors' Addresses

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

Email: leaf.y.yeh@huawei.com

Mohamed Boucadair
France Telecom
Rennes,
France

Email: mohamed.boucadair@orange.com

Ted Lemon
Nominum, Inc
USA

Email: Ted.Lemon@nominum.com

Jie Hu
China Telecom
Beijing,
P. R. China

Email: huj@ctbri.com.cn

