DHC
Internet-Draft
Intended status: Standards Track
Expires: December 17, 2011

E. Bi
S. Manning
M. Wong
Huawei Technologies
June 15, 2011

Option Extensions for DHCPv4
draft-bi-dhc-opextensions-00.txt

Abstract

This document defines a new option that can be used by DHCP servers to exchange with DHCP clients specific security configuration information. This new options also defines a standard parameter format and code so that there will be no ambiguity in interoperating the information being exchanged and that there will be no issues related to interoperability. It also defines some other options that may be used in enterprise networks but not included in [RFC2132].

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 December 18, 2011.

Copyright Notice

Table of Contents

1. Introduction

DHCP provides a framework for passing network configuration information to hosts on a TCP/IP network. Some configuration parameters and control information can be carried in DHCP options which are defined in [RFC2132], [RFC3046], [RFC3118], [RFC4030], etc. When a host that acts as a DHCP client booting up, it can be configured with some security policy, e.g., due to the security concern, all the configuration packets to and from a client must be transported via a secure channel which is established with the server or administrator. Some scenarios that require this kind of secure booting are when DHCP clients are in wireless base stations attaching to a wireless network infrastructure as defined in [3GPP.33.310]. Otherwise, the packets may be dropped by nodes in the network such as a firewall or security gateway. So it is important for the host to obtain a set of security information, which is configured in the DHCP server prior to the establishment of the security tunnel. Currently, some implementations exchange this security information through DHCP vendor-specific options. However, this has the usual limitations of requiring the client and server to understand these vendor specific extensions. Since most of the parameters that make up the security information are common across most clients and servers, having a standardized set of options and procedures would be a huge benefit to interoperability. Some implementations need additional options that are not yet defined in [RFC2132] and therefore create interoperability issues among implementations by different vendors. This document defines a new set of common DHCP options used to exchange the security information and related parameters.

The four newly defined options are as follows:

Option X1: DHCP Security Specific Configuration option

Option X2: Proxy Server option

Option X3: Local Print Server option

Option X4: Local File Server option

2. Terminology used in this document

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

3. DHCP Security Specific Configuration option

A DHCP server can use this option to indicate to the DHCP client specific configuration information, such as the IP address of the security gateway that is used to establish IPsec tunnel within the enterprise network, or multiple IP addresses of the client that are used within the enterprise network. The information contained in the specific configuration area of this option includes one or more attribute values that are assigned by IANA. The information which is contained in these attribute data fields of this option contains the detailed specific configuration information for the DHCP client.

This option may be used wherever DHCP options may be used, as specified in [RFC2131]and [RFC2132]. It is most meaningful in the messages between DHCP client and DHCP server, such as, DHCPOFFER and DHCPACK. The format of the option X1 is as follows:

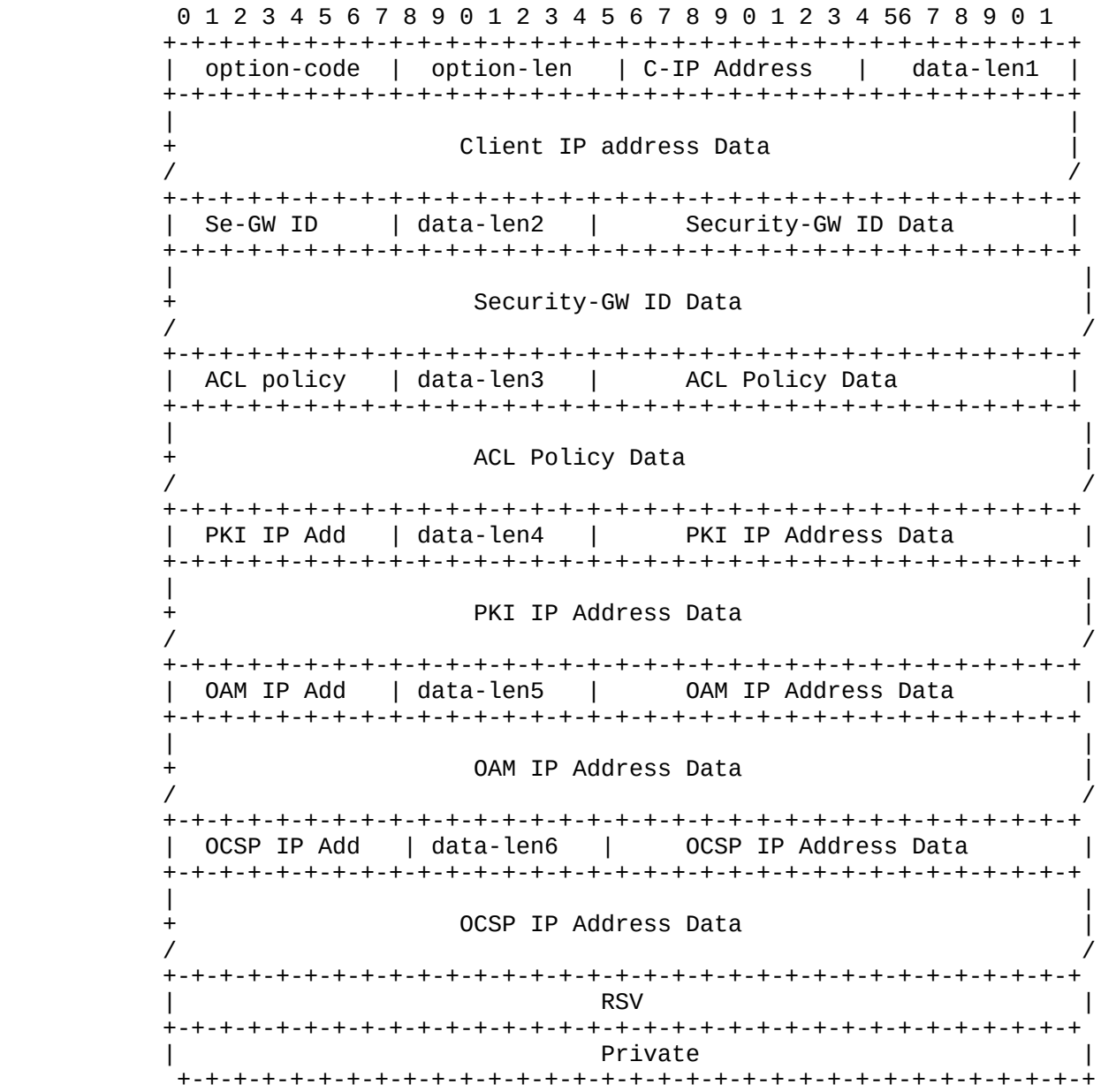


Figure 1. The format of DHCP security specific configuration option

option-code: DHCP security specific configuration option code (TBD).

option-len: Total length of all following option data in octets.

Security-specific attribute N: Security-specific attribute type code (TBD).

If the DHCP client is required to securely boot, the address or FQDN of Security gateway for IPsec establishment is required. Additionally if the security policy dictates, the ACL policy and multiple IP address of the DHCP client for different usage are also required.

In this document, the minimum set of security-specific attribute is specified.

IP address of the DHCP client: it indicates the IP address of the DHCP client. In current specification defined in [RFC2131], the IP address of client is allocated in yiaddr field, but only one address can be obtain. If multiple IP addresses are needed, such as, transport IP, service IP and OM IP or public interface IP address and enterprise network IP address, this option can be used. The security-specific attribute data comprise the different separate items. The definition of the sub-security-specific is listed as follows.

Public interface IP: it indicates the public interface IP address of the DHCP client.

Enterprise IP: it indicates the IP address of the DHCP client used within the enterprise network.Each of the attributes is optional and available according to local policy.

Security-gateway ID: it indicates the security information of the security gateway. If the client is configured with security policy, the value is mandatory to use. Else, it cannot obtain the Security gateway information to establish IPsec tunnel. And it mainly used with IPsec.

IP address: it indicates the IP address of the security gateway.

FQDN: it indicates the FQDN of the security gateway

Either of these two sub-security-specific attributes can be contained according to local policy.

ACL policy: it indicates the ACL policy attribute. And six elements of the ACL policy are contained in the series of sub-security-specific attributes. The six elements which contains IP address, transport port number, protocol, and DSCP. The client will be

configured with an ACL to filter potentially dangerous packets. Only packets that match the ACL parameters are allowed to pass.

RA/CA IP: it indicates the IP address of RA/CA

OMA IP: it indicates the IP address of OAM

OCSP IP: it indicates the IP address of OCSP

For these three attributes, a 3GPP base station as a DHCP client needs to obtain the operator certificate for certificate based authentication as defined in [3GPP.33.310] in wireless network.

RSV: If is not used, it should be set to zero.

Private: It is for private use.

These attributes are for the client, the configuration of all of base stations are the same when initially dispatched from the factory, when the server receive the client ID, the server will know whether the client needs to be configured with security, and then it will send the security information to the client. For the client, it MUST identify the option.

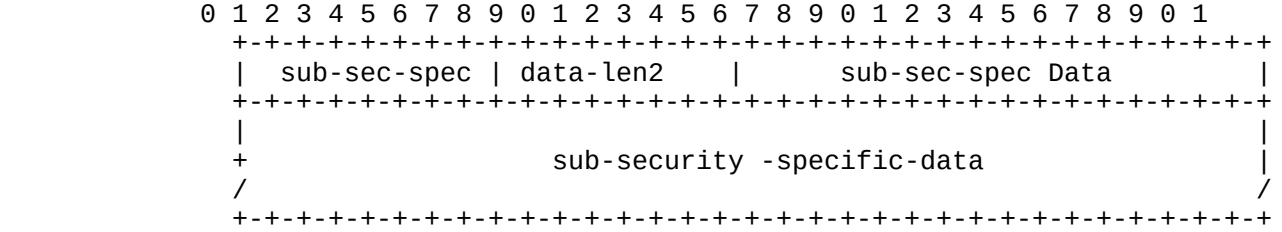


Figure 2. The format of Sub-security-specific option

Sub-security-specific attribute type: type code of the sub-security-specific attribute, i.e., for security-specific attribute3, it includes the source IP address, destination address, source port number, destination port number, protocol and DSCP in order. The data-len is one octet long and specifies the length of the sub-security-specific data.

This option contains the information corresponding to one or more security-specific code number. Multiple instances of this option may be present and must be concatenated in accordance with [RFC3396]. The definition of the information carried in this option is defined

uniformly. The security-specific attribute information indicated the security information type. As the security-specific code is uniform and standard, no ambiguity interpretation can occur. A security-specific code number is unique and only occur once in the option and should be treated independently. This option can also contains one or more encapsulated options that defined in [RFC2132].

DHCP client can request the configuration information from DHCP server by sending DHCP request message. DHCP server allocates the configuration information to DHCP client according to the client ID. i.e., DHCP server can know whether this connecting client needs to be configured security configuration by client ID, which can be carried in option 60 specified in [RFC2132]. If the security configuration is needed, the defined security-specific option will be sent back to the client from DHCP server in DHCPOFFER. If this option is used, different DHCP clients implemented by different vendors have good interoperability. The DHCP server needs only to support one standardized format which reduces complexity and enhances performance.

If the DHCP client is configured with a security policy, all of the attributes listed in the figure MUST be carried in the newly defined option in DHCPDISCOVER or DHCPREQUEST messages. And DHCP server allocate the configuration attribute values according to local policy, the DHCP client MUST has the capability to identify the option.

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

4. Other options can be used for enterprise network

Some other options can be used for enterprise networks.

X.2. Proxy Servers Option

This option specifies a list of IP addresses indicating proxy servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is X2. Its minimum length is 4, and the length MUST be a multiple of 4.

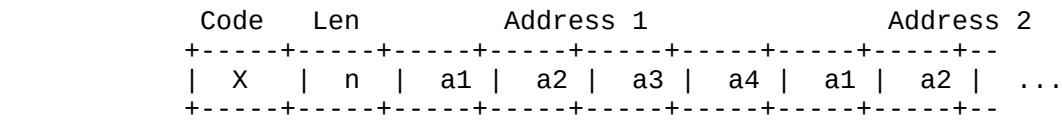


Figure 3. X.2 Proxy Servers Option

This option specifies a list of IP addresses indicating local print servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is X3. Its minimum length is 4, and the length MUST be a multiple of 4.

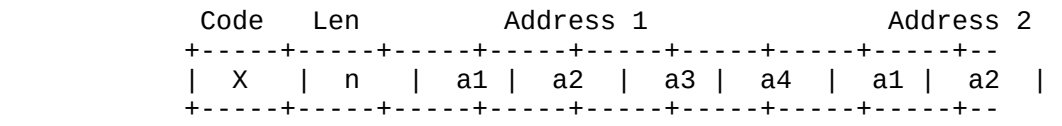


Figure 4. X.3 Local Print Servers Option

This option specifies a list of IP addresses indicating local file servers available to the client. Servers SHOULD be listed in order of preference.

The code for this option is X4. Its minimum length is 4, and the length MUST be a multiple of 4.

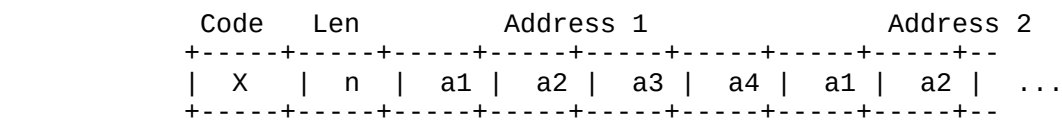


Figure 5. X.4. Local File Servers Option

5. Security Considerations

This document defines a few some options used by DHCP servers and DHCP clients to exchange the additional configuration information.

And, if the additional configuration information is sensitive in nature, consideration needs to be taken on how to protect it.

6. IANA Considerations

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

7. Acknowledgments

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

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.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

[RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.

[RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.

[RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396, November 2002.

[RFC4030] Stapp, M. and T. Lemon, "The Authentication Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Option", RFC 4030, March 2005.

8.2. Informative References

[3GPP.33.310] 3GPP, "Network Domain Security (NDS); Authentication

Framework (AF)", 3GPP TS 33.310 6.2.0, September 2004.

Author’s Address

Emily Bi
Huawei Technologies
Huawei Building, Xinxi Road No.3
Haidian District, Beijing 100085
P. R. China

Phone: +86-10-82881907
Email: bixiaoyu@huawei.com

Serge Manning
Huawei Technologies

Phone: 001-9725435324
Email: serge.manning@huawei.com

Marcus Wong
Huawei Technologies

Phone: 001-908-5413505
Email: mwong@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 12, 2012

Y. Cui
P. Wu
J. Wu
Tsinghua University
July 11, 2011

DHCPv4 Behavior over IP-IP tunnel
draft-cui-softwire-dhcp-over-tunnel-01

Abstract

This document analyzes the scenario in which DHCPv4 interaction is performed over IP-IP tunnel, and proposes methods to keep DHCP working under such situation. The main issue is encapsulation of DHCP packets on server side, and there are both in-protocol and out-of-protocol solutions for this issue. The in-protocol solution is to have DHCP carrying the encapsulation address information, and the out-of-protocol solution is to have the DHCP server keeping track of the address mapping by inspecting DHCP packets.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 12, 2012.

Copyright Notice

Table of Contents

1. Introduction

The DHC protocol[RFC2131] wasn't designed with tunnel environment considerations. However, due to the development of tunnel-based mechanisms, the demand to apply DHCP in tunnel environment arises, especially in the context of IPv6 transition. A typical application scenario is IP-IP Hub and spoke tunnel[RFC4925]. In this type of scenario, IP-IP tunnel is used to provide non-native IP connectivity to hosts, across a heterogenous network. If the non-native IP addresses of the clients are provided by the concentrator side, this address provisioning needs to cross the heterogeneous network, too.

One transition mechanism that requires DHCP over tunnel is documented in [I-D.cui-softwire-host-4over6]. In this mechanism, users in IPv6 network get IPv4 access by IPv4-in-IPv6 tunnel with 4over6 concentrator. Every user employs a public IPv4 address to get full bidirectional IPv4 communication. This IPv4 address is allocated by the ISP over the IPv6 network. The document suggests to achieve this by tunneling DHCPv4 between the 4over6 initiator(DHCPv4 client) and 4over6 concentrator(DHCPv4 server).

Two main flavours of solutions may be considered:

- Use DHCPv6 to provision IPv4-related connectivity, since IPv4 address can be embedded into IPv6 address field. To achieve this mode, dedicated options are needed to convey IPv4-related information, such as IPv4 address of DNS server, NTP server, etc.

- Use DHCPv4 and sustain it in the tunnel environment. Unlike the previous approach where only DHCPv6 is used for both IPv4 and IPv6 connectivity, this approach consists in maintaining the separation between IPv4 and IPv6 connectivity information. It allows to maintain the IPv4 service without requiring major modification of IPv6-related provisioning resources, and perserves DHCP as an IPv4-related information carrier. This document focuses on this flavour.

2. Teminology

This document makes use of the following terms:

- DHCPv4 refers to IPv4 DHCP [RFC2131].
- DHCPv4 client (or client) denotes a node that initiates requests to obtain configuration parameters from one or more DHCP servers [RFC2131].
- DHCPv4 server (or server) refers to a node that responds to requests from DHCP clients [RFC2131].

3. Problem Analysis

The scenario of DHCPv4 over IP-IP tunnel is shown in Figure 1. DHCPv4 client and DHCPv4 server(could be a relay) are separated by an IPv6 or IPv4 network, with no DHCP relay in the middle. DHCP DISCOVER and DHCP REQUEST packets cannot reach the other end since they are broadcast packets; DHCP OFFER and DHCP ACK/NAK packets cannot reach the other end either, when they are broadcast packets or unicast packets forwarded by MAC address. Therefore a tunnel between the client and server is required to build a virtual link. Besides, when the middle network is IPv6-only, all DHCPv4 packets can not go through the network.

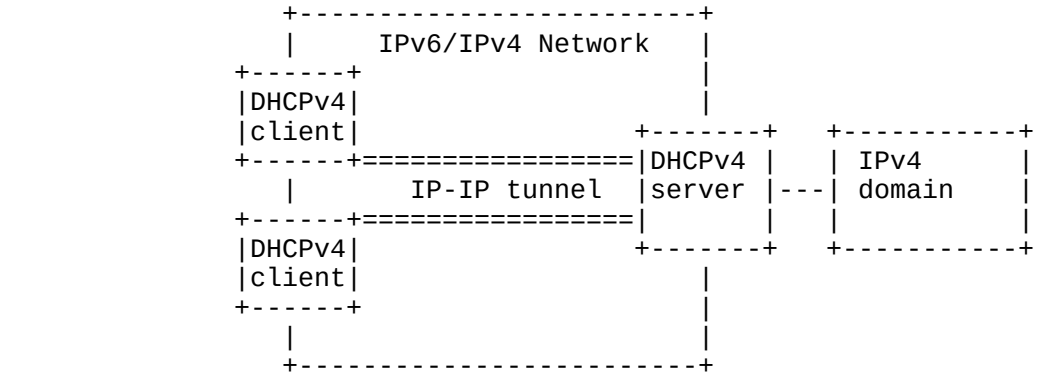


Figure 1 Scenario of DHCPv4 over tunnel

For the above reasons, we need to build the whole DHCP procedure on an IP-IP tunnel. The client(tunnel initiator) and server(tunnel concentrator) will encapsulate the E-IP(External-IP, IPv4) DHCP packets into I-IP(Internal-IP, could be IPv4 or IPv6) before sending them to remote end; the remote end(server or client) will decapsulate the packets to get the original E-IP DHCP packet before handing them to the DHCP process. The encapsulation on the client is natural: the client will use the server's I-IP address as encapsulation destination address, which is usually known beforehand. The problem is the encapsulation on the server. The server serves more than one clients, and it must send every DHCP packet to the right client, each with different I-IP address.

We can see that regular data packet encapsulation on the concentrator faces a similar problem. The solution is to have the concentrator maintaining the mapping between each initiator's E-IP address and I-IP address. When the concentrator performs encapsulation, it will

use the packet’s E-IP destination address to look up the I-IP encapsulation destination address. However, this solution doesn’t apply to DHCP packets, because the address mapping can only be established after the DHCP address allocation, and also because the destination address of DHCP packets can be broadcast address. So we need some extra effort to make the encapsulation of DHCP packets work, i.e., make the concentrator encapsulate each DHCP packet with the I-IP address of the right initiator and hence send it to the right initiator.

4. In-protocol and Out-of-protocol Solutions

So far we've come to two solutions for this problem, one is an in-protocol solution and the other is an out-of-protocol solution. In this version of draft, we describe both of them for further discussion.

4.1. Address mapping with session id

This is an out-of-protocol solution. The basic idea is that the concentrator(server) inspects the incoming DHCP packets, keeps track of the mapping between the DHCP session id and the I-IP address of the packet. When sending out a DHCP packet, the concentrator will use the session id in the packet to look up corresponding I-IP address for encapsulation. Here the session id could be any field in the DHCP packet that can be used to distinguish different clients, such as MAC address, transaction-id, etc. The mapping needs to last for only the lifetime of two-time handshake.

Figure 2 provides an example using MAC as session id. When receiving a DHCPDISCOVER message, the concentrator stores the mapping between the MAC address and I-IP address in encapsulation header. Then the concentrator decapsulates the packet and hands the packet to upper layer. When the upper layer passes down the corresponding DHCPOFFER packet, the concentrator will look up the I-IP address in the mapping table, using the MAC address in the DHCPOFFER packet. This I-IP address will be used as encapsulation destination address. Then the mapping can expire. Similar procedure happens when the concentrator receives a DHCPREQUEST and sends out a DHCPACK.

This method is transparent to the DHCP process. There's no protocol extension required. However, the concentrator need to inspect every encapsulated packet to filter out DHCP packets.

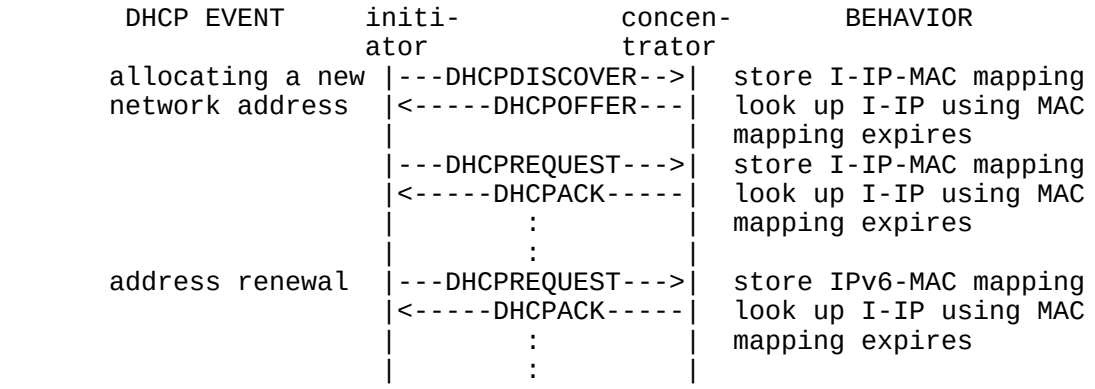


Figure 2 4over6 concentrator: DHCP behavior

4.2. Leveraging Relay Agent Option

Unlike the first solution, the second solution is an in-protocol solution. We can see that what is actually needed to solve this problem is an I-IP encapsulation address for every DHCP packet. We can have the DHCP client to include this information in every DHCP packet it sends out. This document suggests to use the Agent Circuit ID Sub-option in DHCP Relay Agent Information Option (Option 82) [RFC3046] to carry the I-IP address information.

Having the client doing this, the operations on the concentrator can be significantly simplified. The receiving and decapsulating procedure of the DHCP packet can be identical to regular data packet. The DHCP server process will not modify Option 82 in a DHCP packet, and this option will be included in the DHCP reply packet. When the upper layer passes down the DHCP reply packet, the concentrator will look into the packet and find the encapsulation address in Option 82. Then the encapsulation can be done easily.

This method doesn’t need per-packet inspecting when decapsulating packet, and doesn’t need address mapping maintenance, either. However, it’s a " misuse" of Option 82 in some level, since there’s actually no DHCP relay involved. Another possibility is that we can define a new DHCP option for this specific usage if it is necessary.

5. Acknowledgement

The authors would like to thank Alain Durand, Yiu L. Lee, Ted Lemmon and Mohamed Boucadair for their valuable comments on this draft.

6. References

6.1. Normative References

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.

[RFC4925] Li, X., Dawkins, S., Ward, D., and A. Durand, "Softwire Problem Statement", RFC 4925, July 2007.

6.2. Informative References

[I-D.boucadair-dhcpv6-shared-address-option] Boucadair, M., Levis, P., Grimault, J., Savolainen, T., and G. Bajko, "Dynamic Host Configuration Protocol (DHCPv6) Options for Shared IP Addresses Solutions", draft-boucadair-dhcpv6-shared-address-option-01 (work in progress), December 2009.

[I-D.cui-softwire-host-4over6] Cui, Y., Wu, J., Wu, P., Metz, C., Vautrin, O., and Y. Lee, "Public IPv4 over Access IPv6 Network", draft-cui-softwire-host-4over6-06 (work in progress), July 2011.

Authors' Addresses

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

Phone: +86-10-6260-3059
Email: yong@csnet1.cs.tsinghua.edu.cn

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

Phone: +86-10-6278-5822
Email: weapon@csnet1.cs.tsinghua.edu.cn

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

Phone: +86-10-6278-5983
Email: jianping@cernet.edu.cn

6man Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 30, 2011

A. Matsumoto
T. Fujisaki
J. Kato
NTT
T. Chown
University of Southampton

June 28, 2011

Distributing Address Selection Policy using DHCPv6
draft-ietf-6man-addr-select-opt-01.txt

Abstract

RFC 3484 defines default address selection mechanisms for IPv6 that allow nodes to select appropriate address when faced with multiple source and/or destination addresses to choose between. The RFC allowed for the future definition of methods to administratively configure the address selection policy information. This document defines a new DHCPv6 option for such configuration, allowing a site administrator to distribute address selection policy, and thus control the address selection behavior of nodes in their site. While RFC 3484 is in the process of being updated, with a revised default policy table, that table may not suit every scenario, and thus the DHCPv6 option defined in this text may be used to override that policy where desired.

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 December 30, 2011.

1. Introduction

RFC 3484 [RFC3484] describes default algorithms for selecting an address when a node has multiple destination and/or source addresses to choose between by using an address selection policy. In Section 2 of RFC 3484, it is suggested that the default policy table may be administratively configured to suit the specific needs of a site. This text defines a new DHCPv6 option for such configuration.

Some problems have been identified with the default address selection policy detailed in RFC 3484 [RFC5220], and as a result the RFC is in the process of being updated, as per [I-D.ietf-6man-rfc3484-revise]. While this update provides a better default address selection policy, it is unlikely that such a default will suit all scenarios, and thus mechanisms to control the source address selection policy will be necessary. Requirements for those mechanisms are described in [RFC5221], while solutions are discussed in [I-D.ietf-6man-addr-select-sol] and [I-D.ietf-6man-addr-select-considerations]. Those documents have helped shape the improvements in [I-D.ietf-6man-rfc3484-revise] as well as the DHCPv6 option defined here.

1.1. Conventions Used in This Document

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

1.2. Terminology

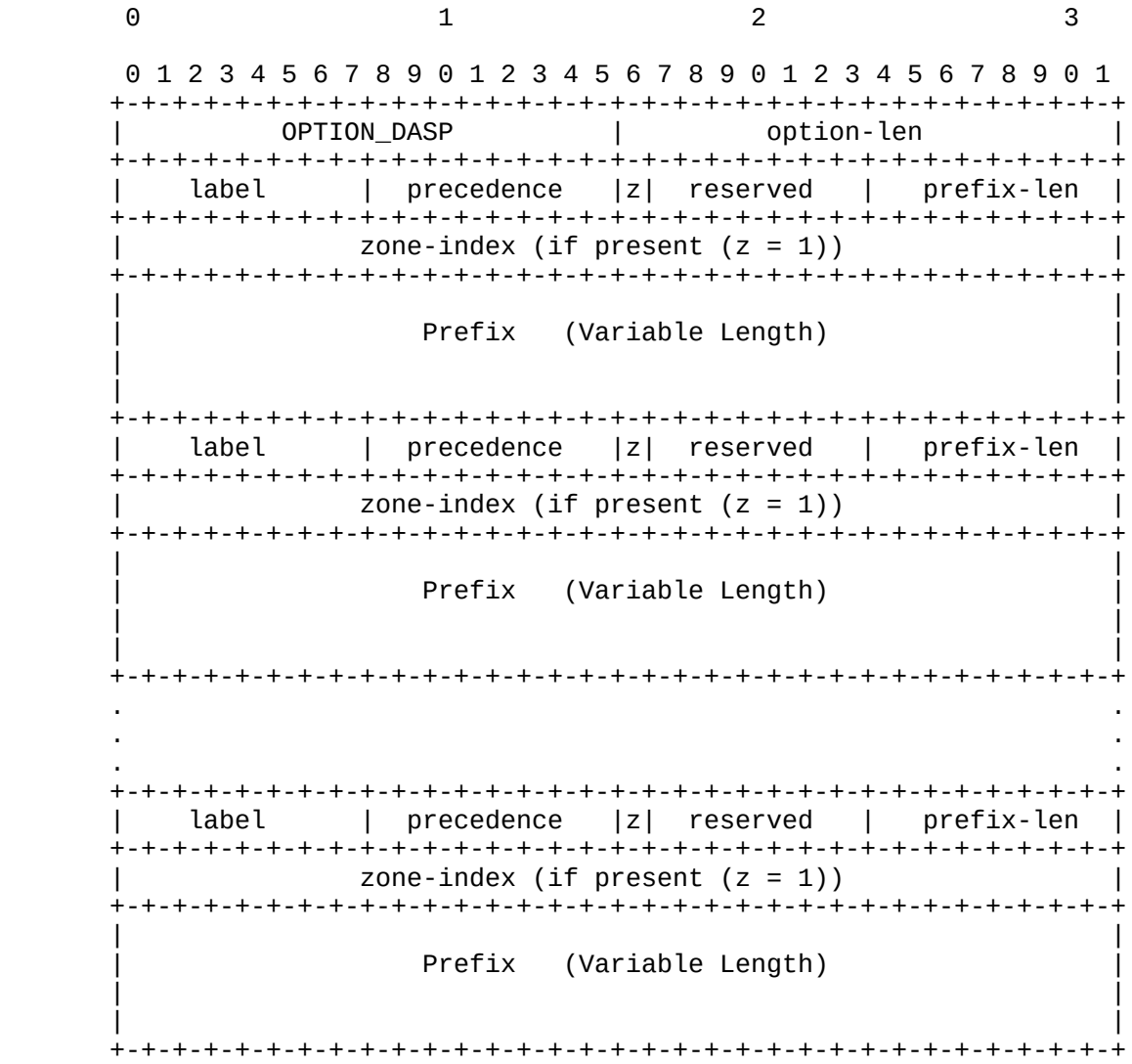
This document uses the terminology defined in [RFC2460] and the DHCPv6 specification defined in [RFC3315]

2. Address Selection Policy Option

The Address Selection Policy Option provides the policy table for address selection rules as described in RFC 3484 and updated in [I-D.ietf-6man-rfc3484-revise].

Each end node is expected to configure its policy table, as described in RFC 3484, using the Address Selection Policy option information as described in the section below on processing the option.

The format of the Address Selection Policy option is given 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_DASP
option-len
label
precedence
z
reserved
prefix-len
zone-index (if present (z = 1))
Prefix (Variable Length)
label
precedence
z
reserved
prefix-len
zone-index (if present (z = 1))
Prefix (Variable Length)
label
precedence
z
reserved
prefix-len
zone-index (if present (z = 1))
Prefix (Variable Length)

[Fig. 1]

Fields:

option-code: OPTION_DASP (TBD)

option-len: The total length of the label fields, precedence fields, zone-index fields, prefix-len fields, and prefix fields in octets.

label: An 8-bit unsigned integer; this value is used to make a combination of source address prefixes and destination address prefixes.

precedence: An 8-bit unsigned integer; this value is used for sorting destination addresses.

z bit: 'zone-index' bit. If z bit is set to 1, 32 bit zone-index value is included right after the "prefix-len" field, and "Prefix" value continues after the "zone-index" field. If z bit is 0, "Prefix" value continues right after the "prefix-len" value.

reserved: 6-bit reserved field. Initialized to zero by sender, and ignored by receiver.

zone-index: If the z-bit is set to 1, this field is inserted between "prefix-len" field and "Prefix" field. The zone-index field is an 32-bit unsigned integer and used to specify zones for scoped addresses. This bit length is defined in RFC3493 [RFC3493] as 'scope ID'.

prefix-len: An 8-bit unsigned integer; the number of leading bits in the prefix that are valid. The value ranges from 0 to 128. The Prefix field is 0, 4, 8, 12, or 16 octets, depending on the length.

Prefix: A variable-length field containing an IP address or the prefix of an IP address. An IPv4-mapped address [RFC4291] must be used to represent an IPv4 address as a prefix value.

3. Appearance of this Option

The Address Selection Policy option MUST NOT appear in any messages other than the following ones: Solicit, Advertise, Request, Renew, Rebind, Information-Request, and Reply.

4. Processing the Address Selection Policy Option

This section describes how to process received Address Selection Policy Options at the DHCPv6 client.

This option's concept is to serve as a hint for a node about how to behave in the network. So, basically, it should be up to the node's administrator how to make use of or even ignore the received policy information.

However, we need to define the default behavior of the receiving node in order to reduce operational complexity.

4.1. Handling the local policy table

RFC3484 defines the default policy for the policy table. Also, a user is usually able to configure the policy table to satisfy his requirement.

The client node SHOULD provide the following choices:

a) It receives distributed policy table, and replaces the existing policy tables with that.
b) It preserves the default policy table, or manually configured policy.

4.2. Processing multiple received policy tables

The policy table is node-global information by its nature. So, the node cannot use multiple received policy tables at the same time.

It should be noted that adopting a received policy table as the node-global information can cause security problems, such as DOS attack, and leak of privacy information.

Moreover, it also should be noted that, when a node is single-homed and has only one upstream line, adopting a received policy table does not degrade the security level.

Under the above assumptions, we specify how to handle multiple received policy tables below.

A node MAY use OPTION_DASP in any of the following two cases:

1: The address selection option is delivered across a secure, trusted channel.
2: The address selection option is not secured, but the node is single-homed.

In other cases the node MUST NOT use OPTION_DASP unless the node is specifically configured to do so.

5. Implementation Considerations

- The value 'label' is passed as an unsigned integer, but there is no special meaning for the value, that is whether it is a large or small number. It is used to select a preferred source address prefix corresponding to a destination address prefix by matching the same label value within the DHCP message. DHCPv6 clients need to convert this label to a representation specified by each implementation (e.g., string).

- Currently, the label and precedence values are defined as 8-bit unsigned integers. In almost all cases, this value will be enough.

- The maximum number of address selection rules that may be conveyed in one DHCPv6 message depends on the prefix length of each rule and the maximum DHCPv6 message size defined in RFC 3315. It is possible to carry over 3,000 rules in one DHCPv6 message (maximum UDP message size), but the usual number would be much smaller, e.g. the default policy table defined in RFC 3484 contains 5 rules.

- Since the number of selection rules could be large, an administrator configuring the policy to be distributed should consider the resulting DHCPv6 message size.

6. Security Considerations

A rogue DHCPv6 server could issue bogus address selection policies to a client. This might lead to incorrect address selection by the client, and the affected packets might be blocked at an outgoing ISP because of ingress filtering. Alternatively, an IPv6 transition mechanism might be preferred over native IPv6, even if it is available.

To guard against such attacks, both DCHP clients and servers SHOULD use DHCP authentication, as described in section 21 of RFC 3315,

"Authentication of DHCP messages."

7. IANA Considerations

IANA is requested to assign option codes to OPTION_DASP from the option-code space as defined in section "DHCPv6 Options" of RFC 3315.

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.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

8.2. Informative References

[I-D.ietf-6man-addr-select-considerations] Chown, T., "Considerations for IPv6 Address Selection Policy Changes", draft-ietf-6man-addr-select-considerations-03 (work in progress), March 2011.

[I-D.ietf-6man-addr-select-sol] Matsumoto, A., Fujisaki, T., and R. Hiromi, "Solution approaches for address-selection problems", draft-ietf-6man-addr-select-sol-03 (work in progress), March 2010.

[I-D.ietf-6man-rfc3484-revise] Matsumoto, A., Kato, J., and T. Fujisaki, "Update to RFC 3484 Default Address Selection for IPv6", draft-ietf-6man-rfc3484-revise-03 (work in progress), June 2011.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

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

RFC 3493, February 2003.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC5220] Matsumoto, A., Fujisaki, T., Hiromi, R., and K. Kanayama, "Problem Statement for Default Address Selection in Multi-Prefix Environments: Operational Issues of RFC 3484 Default Rules", RFC 5220, July 2008.

[RFC5221] Matsumoto, A., Fujisaki, T., Hiromi, R., and K. Kanayama, "Requirements for Address Selection Mechanisms", RFC 5221, July 2008.

Appendix A. Past Discussion

- The 'zone index' value is used to specify a particular zone for scoped addresses. This can be used effectively to control address selection in the site scope (e.g., to tell a node to use a specified source address corresponding to a site-scoped multicast address). However, in some cases such as a link-local scope address, the value specifying one zone is only meaningful locally within that node. There might be some cases where the administrator knows which clients are on the network and wants specific interfaces to be used though. However, in general case, it is hard to use this value.

- Since we got a comment that some implementations use 32-bit integers for zone index value, we extended the bit length of the 'zone index' field. However, as described above, there might be few cases to specify 'zone index' in policy distribution, we defined this field as optional, controlled by a flag.

- There may be some demands to control the use of special address types such as the temporary addresses described in RFC4941 [RFC4941], address assigned by DHCPv6 and so on. (e.g., informing not to use a temporary address when it communicate within the an organization's network). It is possible to indicate the type of addresses using reserved field value.

Authors' Addresses

Arifumi Matsumoto
NTT SI Lab
3-9-11 Midori-Cho
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 3334
Email: arifumi@nttv6.net

Tomohiro Fujisaki
NTT PF Lab
3-9-11 Midori-Cho
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 7351
Email: fujisaki@nttv6.net

Jun-ya Kato
NTT SI Lab
3-9-11 Midori-Cho
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 2939
Email: kato@syce.net

Tim Chown
University of Southampton
Southampton, Hampshire SO17 1BJ
United Kingdom

Email: tjc@ecs.soton.ac.uk

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

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

July 11, 2011

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

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on January 12, 2012.

Table of Contents

1. Introduction

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

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

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

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

1.1. Requirements Language

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

1.2. Terminology

The following terms and acronyms are used in this document:

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

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

2. Protocol Summary

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

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

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

2.1. RELAYFORWARD Message

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

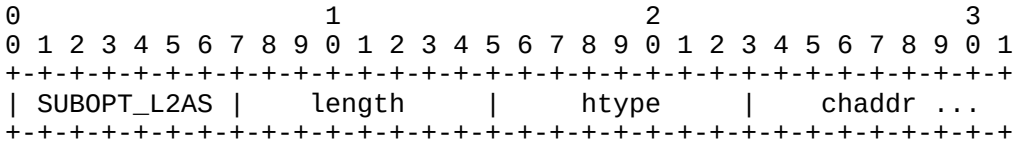
2.2. RELAYREPLY Message

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

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

2.3. Layer Two Address suboption

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



The Layer Two Address suboption has four fields:

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

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

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

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

3. Encoding

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

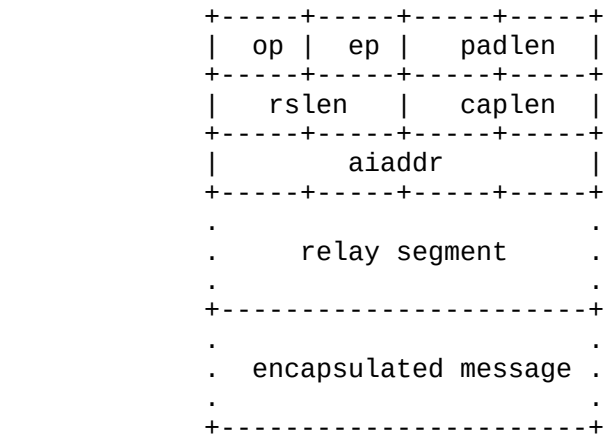
In non-relay DHCP packets, the 'op' field either contains BOOTREQUEST, for any DHCP message from the client to the server, or BOOTREPLY, for any DHCP message from the server to the client.

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

code	meaning
1	BOOTREQUEST
2	BOOTREPLY
3	RELAYFORWARD
4	RELAYREPLY

Values for the 'op' field

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



3.1. The fixed-length header

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

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

The fixed-length header consists of five fields:

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

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

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

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

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

aiaddr Relay agent IP address.

3.2. Relay Segment

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

3.3. Encapsulation Segment

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

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

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

4. DHCP Relay Agent Behavior

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

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

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

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

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

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

4.1. Packet processing

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

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

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

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

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

4.1.1. Packets traveling toward DHCP servers

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

4.1.2. Packets traveling toward DHCP clients

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

4.1.3. Anti-spoofing

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

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

4.2. Constructing RELAYFORWARD messages

4.2.1. Initializing the fixed-length header

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

If the relay agent is not a layer two relay agent

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

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

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

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

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

4.2.2. Initializing the relay segment

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

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

4.2.3. Fixed header settings for RELAYFORWARD messages

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

4.2.4. Fixed header settings for BOOTREQUEST messages

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

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

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

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

4.2.5. Initializing the encapsulation segment

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

4.3. Decapsulating RELAYREPLY messages

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

4.3.1. Processing relay agent suboptions

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

4.3.2. Constructing the decapsulated message

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

4.4. Retransmitting modified messages

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

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

4.4.1. Layer two relay agents

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

4.4.1.1. Constructing the headers

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

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

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

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

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

4.4.1.2. Forwarding the modified packet

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

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

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

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

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

4.4.2. Layer three relay agents

4.4.2.1. Transmitting a decapsulated RELAYREPLY message

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

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

4.4.2.2. Transmitting a decapsulated BOOTREPLY message

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

4.4.2.3. Transmitting other messages

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

5. DHCP Server Behavior

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

5.1. Receiving RELAYFORWARD messages

DHCP servers that implement this specification MUST decapsulate RELAYFORWARD messages.

5.1.1. Decapsulation

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

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

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

5.1.2. Processing of decapsulated suboptions

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

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

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

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

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

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

5.1.3. Address allocation

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

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

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

5.1.3.1. Default link selection algorithm

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

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

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

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

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

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

5.1.3.2. Other link selection algorithms

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

5.2. Responding to RELAYFORWARD messages

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

5.2.1. Constructing a RELAYREPLY encapsulation

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

5.2.1.1. Constructing the relay segments

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

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

5.2.1.2. Constructing the fixed-length header

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

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

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

5.2.2. Transmission of RELAYREPLY messages

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

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

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

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

5.3. Responding to messages other than RELAYFORWARD

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

6. DHCP Client Behavior

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

7. Security Considerations

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

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

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

8. IANA Considerations

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

9. References

9.1. Normative References

[I-D.ietf-dhc-relay-id-suboption] Stapp, M., "The DHCPv4 Relay Agent Identifier Suboption", draft-ietf-dhc-relay-id-suboption-07 (work in progress), July 2009.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC0951] Croft, B. and J. Gilmore, "Bootstrap Protocol", RFC 951, September 1985.

[RFC1542] Wimer, W., "Clarifications and Extensions for the Bootstrap Protocol", RFC 1542, October 1993.

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

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC3011] Waters, G., "The IPv4 Subnet Selection Option for DHCP", RFC 3011, November 2000.

[RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.

[RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.

[RFC3527] Kinnear, K., Stapp, M., Johnson, R., and J. Kumarasamy, "Link Selection sub-option for the Relay Agent Information Option for DHCPv4", RFC 3527, April 2003.

[RFC4030] Stapp, M. and T. Lemon, "The Authentication Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Option", RFC 4030, March 2005.

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

9.2. Informative References

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

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

Authors' Addresses

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

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

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

Email: denghui@chinamobile.com

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

Email: huanglu@chinamobile.com

Dynamic Host Configuration (DHC)
Internet-Draft
Updates: 3633 (if approved)
Intended status: Standards Track
Expires: December 22, 2011

J. Korhonen, Ed.
Nokia Siemens Networks
T. Savolainen
Nokia
S. Krishnan
Ericsson
O. Troan
Cisco Systems, Inc

June 20, 2011

Prefix Exclude Option for DHCPv6-based Prefix Delegation
draft-ietf-dhc-pd-exclude-02.txt

Abstract

This specification defines an optional mechanism to allow exclusion of one specific prefix from a delegated prefix set when using DHCPv6-based prefix delegation. The new mechanism updates RFC 3633.

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 December 22, 2011.

Copyright Notice

Table of Contents

1. Introduction

This specification defines an optional mechanism and the related DHCPv6 option to allow exclusion of one specific prefix from a delegated prefix set when using DHCPv6-based prefix delegation.

The prefix exclusion mechanism is targeted to deployments where DHCPv6-based prefix delegation is used but a single aggregatable route/prefix has to represent one customer, instead of using one prefix for the link between the delegating router and the requesting router and another prefix for the customer network. The mechanism defined in this specification allows a delegating router to use a prefix out of the delegated prefix set on the link through which it exchanges DHCPv6 messages with the requesting router.

2. Requirements and 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. Problem Background

DHCPv6 Prefix Delegation (DHCPv6-PD) [RFC3633] has an explicit limitation described in Section 12.1 of [RFC3633] that a prefix delegated to a requesting router cannot be used by the delegating router. This restriction implies that the delegating router will have two (non aggregatable) routes towards a customer, one for the link between the requesting router and the delegating router, and one for the customer site behind the requesting router.

There are architectures and link models, where a host (e.g. a mobile router, also acting as a requesting router) always has a single (/64) prefix configured on its uplink interface and the delegating router is also requesting router’s first hop router. Furthermore, it may be required that the prefix configured on the uplink interface has to be aggregatable with the delegated prefixes. This introduces a problem in how to use DHCPv6-PD together with stateless [RFC4862] or stateful [RFC3315] address autoconfiguration on a link, where the /64 advertised on the link is also part of the prefix delegated (e.g /56) to the requesting router.

4. Solution

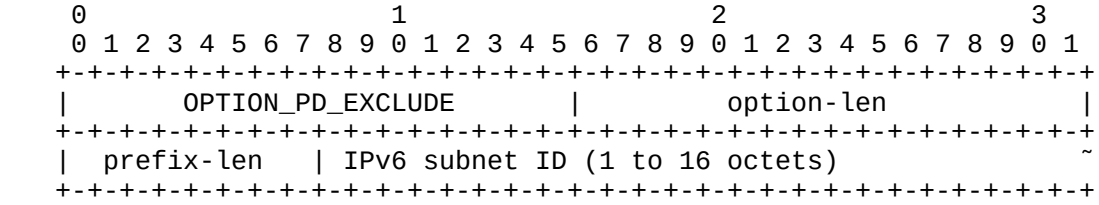
4.1. Prefix Delegation with Excluded Prefixes

This specification defines a new DHCPv6 option, OPTION_PD_EXCLUDE (TBD1), that is used to exclude exactly one prefix from a delegated prefix. The OPTION_PD_EXCLUDE is included in the OPTION_IAPREFIX IAprefix-options field. There can be at most one OPTION_PD_EXCLUDE option in one OPTION_IAPREFIX option. The OPTION_PD_EXCLUDE option allows prefix delegation where a requesting router is delegated a prefix (e.g. /56) and the delegating router uses one prefix (e.g. /64) on the link through which it exchanges DHCPv6 messages with the requesting router with a prefix out of the same delegated prefix set.

A requesting router includes an OPTION_ORO option with the OPTION_PD_EXCLUDE option code in a Solicit, Request, Renew, Rebind or Confirm message to inform the delegating router about the support for the prefix delegation functionality defined in this specification. A delegating router may include the OPTION_PD_EXCLUDE option code in an OPTION_ORO option in a Reconfigure message for indicating that the requesting router should request OPTION_PD_EXCLUDE from the delegating router.

The delegating router includes the prefix in the OPTION_PD_EXCLUDE option that is excluded from the delegated prefix set. The requesting router MUST NOT assign the excluded prefix to any of its downstream interfaces.

4.2. Prefix Exclude Option



Prefix Exclude Option

- option-code: OPTION_PD_EXCLUDE (TBD1).
- option-len: 1 + length of IPv6 subnet ID in octets. A valid option-len is between 2 and 17.
- prefix-len: The length of the excluded prefix in bits. The prefix-len MUST be between 'OPTION_IAPREFIX prefix-length'+1 and 128.

o IPv6 subnet ID: A variable length IPv6 subnet ID up to 128 bits.

The IPv6 subnet ID contains prefix-len minus 'OPTION_IAPREFIX prefix-length' bits extracted from the excluded prefix starting from the bit position 'OPTION_IAPREFIX prefix-length'. The extracted subnet ID MUST be left shifted to start from a full octet boundary, i.e. left shift of 'OPTION_IAPREFIX prefix-length' mod 8 bits. The subnet ID MUST be zero padded to the next full octet boundary.

The encoding of the IPv6 subnet ID can be expressed in a C-like pseudo code as shown below:

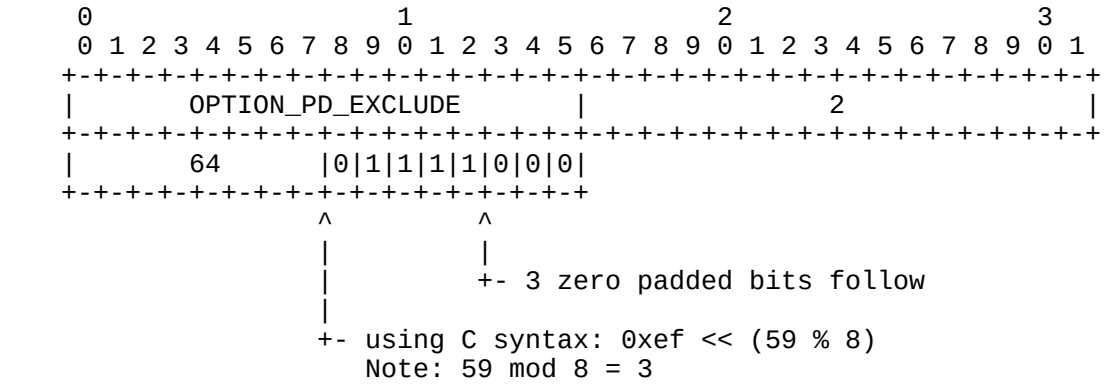
```
uint128_t p1;           // the delegated IPv6 prefix
uint128_t p2;           // the excluded IPv6 prefix
uint16_t a;             // the OPTION_IAPREFIX prefix-length
uint8_t b;              // the excluded IPv6 prefix length
uint8_t s;

// sanity checks

s = 128-a;              // size of non-prefix bits
assert(b>a);            // b must be at least a+1
assert(p1>>s == p2>>s); // p1 and p2 must share a common
                        // prefix of 'a' bits

// calculate the option content

uint16_t c = b-a-1;     // the IPv6_subnet_ID_length-1 in bits
uint16_t d = (c/8)+1;   // the IPv6_subnet_ID_length in octets
uint128_t p = p2<<a;    // p is the IPv6 subnet ID that has the
                        // common p1 prefix left shifted out to
                        // a full octet boundary (trailing bits
                        // are zeroed)

// populate the option

uint8_t* id = &OPTION_PD_EXCLUDE.IPv6_subnet_ID;
OPTION_PD_EXCLUDE.option_len = d+1;
OPTION_PD_EXCLUDE.prefix_len = b;

while (d-- > 0) {
  *id++ = p>>120;
  p <<= 8;
}
```

The OPTION_PD_EXCLUDE option MUST only be included in the OPTION_IAPREFIX IAprefix-options [RFC3633] field.

Any prefix excluded from the delegated prefix MUST be contained in OPTION_PD_EXCLUDE options within the corresponding OPTION_IAPREFIX.

The prefix included in the OPTION_PD_EXCLUDE option share the same preferred-lifetime and valid-lifetime as the delegated prefix in the encapsulating OPTION_IAPREFIX option.

The prefix in the OPTION_PD_EXCLUDE option MUST be part of the delegated prefix in the OPTION_IAPREFIX. For example, the requesting router has earlier been assigned a 2001:db8:dead:beef::/64 prefix by the delegating router, and the delegated prefix in the OPTION_IAPREFIX is 2001:db8:dead:bee0::/59. In this case, 2001:db8:dead:beef::/64 is a valid prefix to be used in the OPTION_PD_EXCLUDE option. The OPTION_PD_EXCLUDE option would be encoded as follows:



5. Delegating Router Solicitation

The requesting router locates and selects a delegating router in the same way as described in Section 11 [RFC3633]. This specification only describes the additional steps required by the use of OPTION_PD_EXCLUDE option.

5.1. Requesting Router

If the requesting router implements the solution described in Section 4.1 then the requesting router SHOULD include the OPTION_PD_EXCLUDE option code in the OPTION_ORO option in Solicit messages.

Once receiving Advertise message, the requesting router uses the prefix(es) received in OPTION_PD_EXCLUDE in addition to the advertised prefixes to choose the delegating router to respond to. If Advertise message did not include OPTION_PD_EXCLUDE option, then

the requesting router MUST fall back to normal [RFC3633] Section 11.1 behavior.

5.2. Delegating Router

If the OPTION_ORO option in the Solicit message includes the OPTION_PD_EXCLUDE option code, then the delegating router knows that the requesting router supports the solution defined in this specification. If the Solicit message also contains an IA_PD option, the delegating router can delegate to the requesting router a prefix which includes the prefix already assigned to the requesting router's uplink interface. The delegating router includes the prefix originally or to be assigned to the requesting router in the OPTION_PD_EXCLUDE option within the OPTION_IAPREFIX IAprefix-option in the Advertise message.

If the OPTION_ORO option in the Solicit message does not include the OPTION_PD_EXCLUDE option code, then the delegating router MUST fall back to normal [RFC3633] Section 11.2 behavior.

If the OPTION_ORO option in the Solicit message includes the OPTION_PD_EXCLUDE 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 normal [RFC3633] behavior.

6. Requesting Router Initiated Prefix Delegation

The procedures described in the following sections are aligned with Section 12 of [RFC3633]. In this specification we only describe the additional steps required by the use of OPTION_PD_EXCLUDE option.

6.1. Requesting Router

The requesting router behavior regarding the use of the OPTION_PD_EXCLUDE option is more or less identical to step described in Section 5.1. The only difference really is different used DHCPv6 messages. The requesting router SHOULD include the OPTION_PD_EXCLUDE option code in the OPTION_ORO option in DHCPv6 messages as described in Section 22.7 of [RFC3315].

The requesting router uses a Release message to return the delegated prefix(es) to a delegating router. The prefix(es) to be released MUST be included in the IA_PDs along with the excluded prefix included in the OPTION_PD_EXCLUDE option. The requesting router MUST NOT use the OPTION_PD_EXCLUDE option to introduce additional excluded prefix in the Release message that it originally got a valid binding

for.

The requesting router must create sink routes for the delegated prefixes minus the excluded prefixes. This may be done by creating sink routes for delegated prefixes and more specific routes for the excluded prefixes.

6.2. Delegating Router

The delegating router behavior regarding the use of the OPTION_PD_EXCLUDE option is more or less identical to step described in Section 5.2. The only difference really is DHCPv6 messages used to carry the OPTION_PD_EXCLUDE option.

The delegating router may mark any prefix(es) in IA_PD Prefix options in a Release message from a requesting router as 'available' excluding the prefix included in the OPTION_PD_EXCLUDE options. If the Release message contains 'new' excluded prefix in any OPTION_PD_EXCLUDE option, the delegating router MUST send a Reply message with Status Code set to NoBinding for that IA_PD option.

7. Security Considerations

Security considerations in DHCPv6 are described in Section 23 of [RFC3315], and for DHCPv6 Prefix Delegation in Section 12 of [RFC3633].

8. IANA Considerations

A new DHCPv6 Option Code is reserved from DHCPv6 registry for DHCP Option Codes.

OPTION_PD_EXCLUDE is set to TBD1

9. Acknowledgements

Authors would like to thank Ralph Droms, Frank Brockners, Ted Lemon, Julien Laganier, Fredrik Garneij, Sri Gundavelli, Mikael Abrahamsson, Behcet Sarikaya, Jyrki Soini, Deng Hui, Stephen Jacob and Tomek Mrugalski for their valuable comments and discussions.

10. References

10.1. Normative References

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

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

10.2. Informative References

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

Authors' Addresses

Jouni Korhonen (editor)
Nokia Siemens Networks
Linnoitustie 6
FI-02600 Espoo
Finland

Email: jouni.nospam@gmail.com

Teemu Savolainen
Nokia
Hermiankatu 12 D
FI-33720 Tampere
Finland

Email: teemu.savolainen@nokia.com

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

Email: suresh.krishnan@ericsson.com

Ole Troan
Cisco Systems, Inc
Oslo
Norway

Email: ot@cisco.com

DHC Working Group
Internet Draft
Intended status: Standards Track
Expires: December 22, 2011

Sheng Jiang
Huawei Technologies Co., Ltd
Sean Shen
CNNIC

June 16, 2011

Secure DHCPv6 Using CGAs
draft-ietf-dhc-secure-dhcpv6-03.txt

Status of this Memo

Copyright Notice

Abstract

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) enables DHCP servers to pass configuration parameters. It offers configuration flexibility. If not secured, DHCPv6 is vulnerable to various attacks, particularly fake attack. This document analyzes the security issues of DHCPv6 and specifies security mechanisms, mainly using CGAs.

Table of Contents

1. Introduction

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

The requirements of using CGA to secure DHCPv6 have been introduced in [I-D.draft-ietf-csi-dhcpv6-cga-ps]. This document analyzes the security issues of DHCPv6 in more details. This document is aiming to provide mechanisms for improving the security of DHCPv6, thus the address of a DHCP message sender, which can be a DHCP server, a reply agent or a client, is able to be verified by a receiver. It improves communication security of DHCPv6 interaction. The security mechanisms specified in this document is mainly based on the Cryptographically Generated Addresses (CGA [RFC3972]).

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

2. Terminology

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

3. Security Overview of DHCPv6

DHCPv6 is a client/server protocol that provides managed and stateful configuration of devices. It enables DHCPv6 server to auto-configure relevant network parameters on clients through the DHCPv6 message exchanging mechanisms. In the basic DHCPv6 specifications [RFC3315], security of DHCPv6 message can be improved in a few aspects.

In the basic DHCPv6 specifications, regular IPv6 addresses are used. It is possible for a malicious attacker to use a fake address to spoof or launch an attack.

"One attack specific to a DHCP client is the establishment of a malicious server with the intent of providing incorrect configuration information to the client. The motivation for doing so may be to mount a ’man in the middle’ attack that causes the client to communicate with a malicious server instead of a valid server for some service such as DNS or NTP. The malicious server may also mount a denial of service attack through mis-configuration of the client

that causes all network communication from the client to fail." [RFC3315]

"A DHCP client may also be subject to attack through the receipt of a Reconfigure message from a malicious server that causes the client to obtain incorrect configuration information from that server." [RFC3315]

Fake servers can also provide clients with partially correct information that allows the attacker to route traffic through certain host where critical information can be collected. This becomes important to detect and prevent when encrypted traffic is allowed to pass through firewalls. Clients can be configured with bogus data, so that they will assume that the network is down.

Once servers start updating DNS and other directory services, attackers may spoof DHCP servers to register incorrect information in those services.

Another possible attack is that attackers may be able to gain unauthorized access to some resources, such as network access.

The basic DHCPv6 specifications achieve message origin authentication and message integrity via an authentication option with a symmetric key pair. For the key of the hash function, there are two key management mechanisms. Firstly, the key management is out of band, usually manual, i.e. operators set up key database for both server and client before running DHCPv6. Usually multiple keys are deployed once a time and key id is used to specify which key is used. Secondly, a DHCPv6 server sends a reconfigure key to the client in the initial exchange of DHCPv6 messages for future use, in this case security is not guaranteed because this key is transmitted in plaintext. In either way, the security of key itself is in question mark.

Communication between a server and a relay agent, and communication between relay agents, can be secured through the use of IPSec, as described in section 21.1 in [RFC3315]. However, IPSec is quite complicated. A simpler security mechanism may have better deploy ability. Furthermore, the manual configuration and static keys are potential issue makers. Relay agents MAY require other security mechanisms besides IPSec.

4. Secure DHCPv6 Overview

To solve the abovementioned security issues, we introduce CGAs into DHCPv6. CGAs are introduced in [RFC3972]. "CGAs are IPv6 addresses for which the interface identifier is generated by computing a cryptographic one-way hash function from a public key and auxiliary parameters. The binding between the public key and the address can be

verified by re-computing the hash value and by comparing the hash with the interface identifier. Messages sent from an IPv6 address can be protected by attaching the public key and auxiliary parameters and by signing the message with the corresponding private key. The protection works without a certification authority or any security infrastructure."

In this document, a CGA option with an address ownership proof mechanism and a signature option with a corresponding verification mechanism are introduced. A DHCPv6 message (from either a server, a relay agent or a client) with a CGA as source address, can carry the CGA Parameters data structure and a digital signature. The receiver of this DHCPv6 message can verify both the CGA and signature, then process the payload of the DHCPv6 message only if the validation is successful.

With them, the receiver of a DHCP message can verify the sender address of the DHCP message, which improves communication security of DHCP messages. By using the signature option, the verification of data integrity and replay protections can also be achieved without the authentication option.

This documentation focuses on using CGAs to secure the DHCPv6 protocol. It assumes the sender, which uses CGAs, has self-generated or been configured CGAs. The CGA configuration in the DHCPv6 network is out of scope and specified in [I-D.draft-ietf-dhc-cga-config-dhcpv6].

In the relay scenarios, because relay agent restructures the DHCPv6 messages, a receiver would not find the sender’s source CGA address in the DHCPv6 message header. In the client-relay-server scenarios, "the relay agent copies the source address from the header of the IP datagram in which the message was received to the peer-address field of the Relay-forward message" [RFC3315]. The receiver, a DHCPv6 server, can find the sender’s source CGA address in the peer-address field for CGA verification. In the server-relay-client scenarios, a DHCP server knows a client is behind relay(s) if it receives a Relay-forward DHCPv6 message. Then it will reply a Relay-reply message with the server’s source CGA address being carried in the server DUID, which is in the payload. In this way, the receiver, a DHCPv6 client can get the server’s source CGA address for CGA verification. The server DUID is also protected by CGA.

4.1. New Components

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

- CGAs are used to make sure that the sender of a DHCPv6 message is the "owner" of the claimed address. A public-private key

pair has been generated by a node itself before it can claim an address. A new DHCPv6 option, the CGA Parameter Option, is used to carry the public key and associated parameters.

- Public key signatures protect the integrity of the messages and authenticate the identity of their sender. The authority of a public key is established through the address ownership proof mechanism, by using CGAs.

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

4.2. Support for algorithm agility

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

Following the approach recommended by [RFC4270] and [NewHash], our analysis shows none of these attacks are currently doable. However, these attacks indicate the possibility of future real-world attacks. Therefore, we have to take into account that future attacks will be improved and provide a support for multiple hash algorithms. Our mechanisms, in this document, support not only hash algorithm agility but also signature algorithm agility.

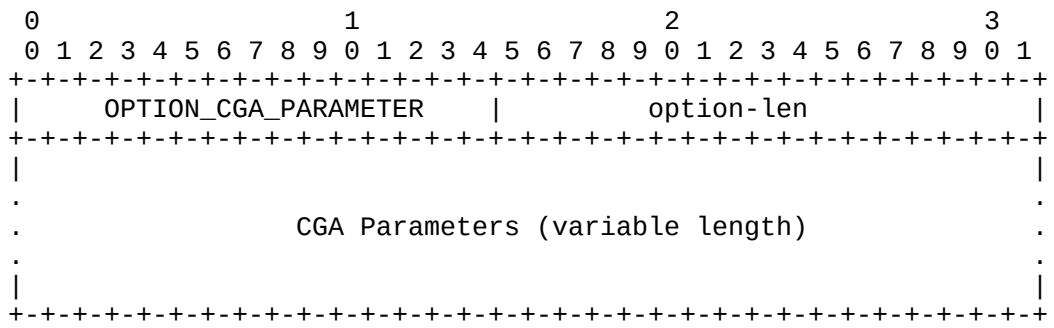
The support for hash agility within CGAs has been defined in [RFC4982]. The usage of CGAs in this document SHOULD also obey [RFC4982], too.

5. Extensions for Secure DHCPv6

This section extends DHCPv6. Two new options and a new DUID type have been defined. The new options MUST be supported, if the node has been configured to use Secure DHCPv6. The new DUID type MUST be supported in the relay scenarios.

5.1. CGA Parameter Option

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



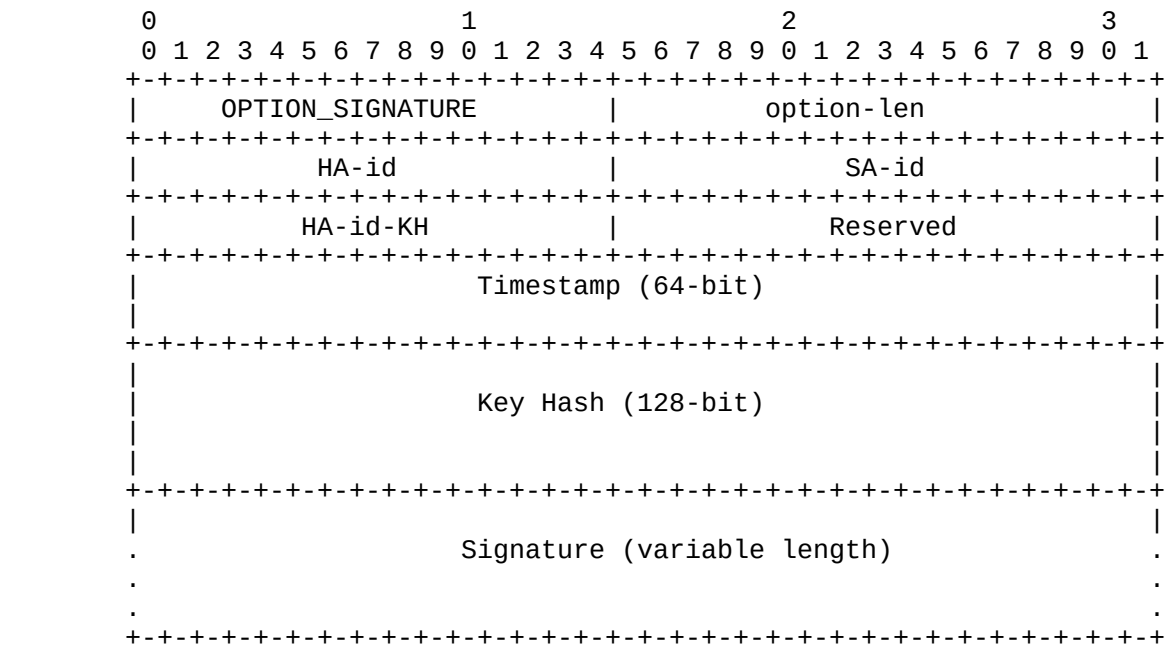
option-code OPTION_CGA_PARAMETER (TBA1).

option-len Length of CGA Parameters in octets.

CGA Parameters A variable-length field containing the CGA Parameters data structure described in Section 4 of [RFC3972]. This specification requires that the public key found from the CGA Parameters field in the CGA option MUST be that referred by the Key Hash field in the Signature option. Packets received with two different keys MUST be silently discarded. Note that a future extension MAY provide a mechanism allowing the owner of an address and the signer to be different parties.

5.2. Signature Option

The Signature option allows public key-based signatures to be attached to a DHCPv6 message. The Signature option COULD be any place within the DHCPv6 message. It protects all the DHCPv6 header and options before it. Any options after the Signature option can be processed, but it should be noticed that they are not protected by this Signature option. The format of the Signature option is described as follows:



option-code OPTION_SIGNATURE (TBA2).

option-len 32 + Length of signature field in octets.

HA-id Hash Algorithm id. The hash algorithm is used for computing the signature result. RSA signature [RSA] with SHA-1 [sha-1] is adopted. In order to provide hash algorithm agility, SHA-1 is assigned an initial value 0x0000 in this document.

SA-id Signature Algorithm id. The signature algorithm is used for computing the signature result. RSA signature with RSASSA-PKCS1-v1_5 algorithm is adopted. In order to provide algorithm agility, RSASSA_PKCS1-v1_5 is assigned an initial value 0x0000 in this document.

HA-id-KH Hash Algorithm id for Key Hash. Hash algorithm used for producing the Key Hash field in the Signature option. SHA-1 is adopted. In order to provide hash algorithm agility, SHA-1 is assigned an initial value 0x0000 in this document.

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

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

Key Hash — A 128-bit field containing the most significant (leftmost) 128 bits of a SHA-1 hash of the public key used for constructing the signature. The SHA-1 hash is taken over the presentation used in the Public Key field of the CGA Parameters data structure carried in the CGA option. Its purpose is to associate the signature to a particular key known by the receiver. Such a key can either be stored in the certificate cache of the receiver or be received in the CGA option in the same message.

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

1. The 128-bit CGA Message Type tag value for Secure DHCPv6, 0x81be a1eb 0021 ce7e caa9 4090 0665 d2e0 02c2. (The tag value has been generated randomly by the editor of this specification.).

2. The 128-bit Source IPv6 Address.

3. The 128-bit Destination IPv6 Address.

4. The DHCPv6 message header.

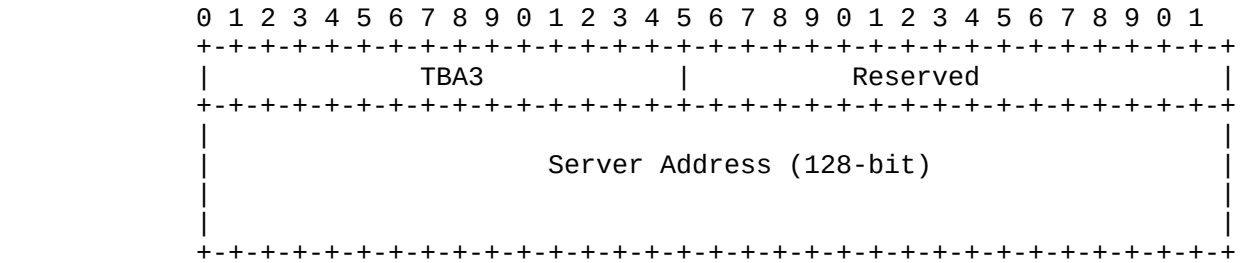
5. All DHCPv6 options except for the Signature option and the Authentication Option.

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

5.3. DUID-SA Type

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

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



Type-code DUID-SA Type (TBA3)

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

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

In the secure DHCPv6 solution, the Server Address field of DUID-SA, which is the IPv6 address of the DHCPv6 server, MUST be a CGA.

In the secure DHCPv6 solution, all the payloads, including DUID-SA, are protected by signature option by the definition of section 5.1 and 5.2.

6. Processing Rules and Behaviors

6.1. Processing Rules of Sender

A DHCPv6 node, which could be a server, relay agent or client, can be configured to send Secure DHCPv6 messages only if CGAs have been configured on it.

The node MUST record the following configuration information:

CGA parameters Any information required to construct CGAs, as described in [RFC3972].

Keypair A public-private key pair. The public key used for constructing the signature MUST be the same in CGA parameters.

CGA flag A flag that indicates whether CGA is used or not.

If a node has been configured to use Secure DHCPv6, the node MUST send a DHCPv6 message using a CGA, which be constructed as specified in Section 4 of [RFC3972], as the source address unless they are sent with the unspecified source address. This DHCPv6 message MUST be signed by the private key of the sender. In the message, both the CGA option and the Signature option MUST be present. The CGA Parameter field in the CGA option is filled according to the rules presented above and in [RFC3972]. The public key in the field is taken from the configuration used to generate the CGA, typically from a data structure associated with the source address. The Signature option MUST be constructed as explained in Section 5.2 and be the last DHCPv6 option.

In relay scenario, a DHCPv6 server MUST include an OPTION_SERVERID [RFC3315] in Relay-reply message and put its CGA in the Server Address field of the DUID in the OPTION_SERVERID. The CGA of DHCPv6 server will not lose during relaying so that the client can verify CGA address and signature.

6.2. Processing Rules of Receiver

The node that supports the verification of the Secure DHCPv6 messages MUST record the following configuration information:

Minbits The minimum acceptable key length for public keys used in the generation of CGAs. The default SHOULD be 1024 bits. Implementations MAY also set an upper limit for the amount of computation needed when verifying packets that use these security associations. Any implementation SHOULD follow prudent cryptographic practice in determining the appropriate key lengths.

On a node that has been configured to use Secure DHCPv6, DHCPv6 message without either the CGA option or the Signature option MUST be treated as unsecured. Note the Secure DHCPv6 nodes MAY simply discard the unsecured messages.

The receiving node MUST verify the source CGA address of the DHCPv6 message by using the public key of the DHCPv6 message sender, CGA Parameters and the algorithm described in Section 5 of [RFC3972]. The inputs to the algorithm are the source address, as used in IP header, and the CGA Parameters field.

If the CGA verification is successful, the recipient proceeds with a more time-consuming cryptographic check of the signature. Note that

even if the CGA verification succeeds, no claims about the validity of the use can be made until the signature has been checked.

The processing on the receiving node also includes the verification on signed data by using the public key of the DHCPv6 message sender. The receiving node MUST verify the Signature option as follows: the Key Hash field MUST indicate the use of a known public key, either one learned from a preceding CGA option in the same message, or one known by other means. The signature field verification MUST show that the signature has been calculated as specified in the previous section.

Only the messages that get through both CGA and signature verifications are accepted as secured DHCPv6 messages and continue to be handled for their contained DHCPv6 options.

Messages that do not pass all the above tests MUST be silently discarded if the host has been configured to accept only secured DHCPv6 messages. The messages MAY be accepted if the host has been configured to accept both secured and unsecured messages but MUST be treated as an unsecured message. The receiver MAY also otherwise silently discard packets.

In the relay scenarios, a DHCPv6 server obtains the CGA of a client from the peer address field in the Relay-forward message. A DHCPv6 client obtains the CGA of a server from the Server Address field of the DUID in the OPTION_SERVERID.

6.3. Processing Rules of Relay Agent

To support secure DHCPv6, Relay Agents follow the same processing rules defined in [RFC3315].

By current definition: "The relay agent copies the source address from the IP datagram in which the message was received from the client into the peer-address field in the Relay-forward message". The CGA of a client will not lose during relaying.

A relay will not change the OPTION_SERVERID when processing Relay-reply message from a DHCPv6 server, CGA of the DHCPv6 server will not lose.

7. Security Considerations

This document provides new security features to the DHCPv6 protocol.

Using CGA as source addresses of DHCPv6 servers, relays or, also in DHCPv6 message exchanging provides the source address ownership verification and data integrity protection.

Without other pre-configured security mechanism, like pre-notified DHCPv6 server address, using host-based CGA by DHCPv6 servers could not prevent attacks claiming to be a DHCPv6 server. Furthermore, CGAs of DHCPv6 servers may be pre-notified to hosts. Then, hosts may decline the DHCPv6 messages from other servers, which may be fake servers. But in this case the address will be fixed. It may increase the vulnerability to, e.g., brute force attacks. The pre-notification operation also needs to be protected, which is out of scope.

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

Downgrade attacks cannot be avoided if nodes are configured to accept both secured and unsecured messages. A future specification MAY provide a mechanism on how to treat unsecured DHCPv6 messages. One simple solution MAY be that Secure DHCPv6 is mandated on all servers, reply agents and clients if a certain link has been deployed Secure DHCPv6.

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

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

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

8. IANA Considerations

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

The CGA Parameter Option (TBA1), described in Section 5.1.

The Signature Option (TBA2), described in Section 5.2.

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

The DUID-SA (TBA3), described in Section 5.3.

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

Hash Algorithm id (HA-id). The values in this name space are 16-bit unsigned integers. The following initial values are assigned for HA-id in this document:

Name	Value	RFCs
SHA-1	0x0000	this document

Signature Algorithm id (SA-id). The values in this name space are 16-bit unsigned integers. The following initial values are assigned for SA-id in this document:

Name	Value	RFCs
SHA-1	0x0000	this document

Hash Algorithm id for Key Hash (HA-id-KH). The values in this name space are 16-bit unsigned integers. The following initial values are assigned for HA-id-KH in this document:

Name	Value	RFCs
RSASSA-PKCS1-v1_5	0x0000	this document

This document defines a new 128-bit value under the CGA Message Type [RFC3972] namespace, 0x81be a1eb 0021 ce7e caa9 4090 0665 d2e0 02c2.

9. Acknowledgments

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

10. References

10.1. Normative References

[RFC3315] R. Droms, et al., "Dynamic Host Configure Protocol for IPv6", RFC3315, July 2003.

[RFC3972] T. Aura, "Cryptographically Generated Address", RFC3972, March 2005.

[RFC4982] M. Bagnulo, J. Arkko, "Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs)", RFC4982, July 2007.

[RFC5905] D. Mills, J. Martin, Ed., J. Burbank and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

10.2. Informative References

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

[RFC4270] Hoffman, P. and B. Schneier, "Attacks on Cryptographic Hashes in Internet Protocols", RFC 4270, November 2005.

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

[RFC6273] A. Kukec, S. Krishnan and S. Jiang "The Secure Neighbor Discovery (SEND) Hash Threat Analysis", RFC 6274, June 2011.

[NewHash] S.Bellovin and E. Rescorla, "Deploying a New Hash Algorithm", November 2005.

[I-D.draft-ietf-dhc-cga-config-dhcpv6] S. Jiang and S. Xia, "Configuring Cryptographically Generated Addresses (CGA) using DHCPv6", draft-ietf-dhc-cga-config-dhcpv6, working in progress, April 2011.

[I-D.draft-ietf-csi-dhcpv6-cga-ps] S. Jiang, S. Shen and T. Chown, "DHCPv6 and CGA Interaction: Problem Statement", draft-ietf-csi-dhcpv6-cga-ps, work in progress, May 2011.

[RSA] RSA Laboratories, "RSA Encryption Standard, Version 2.1", PKCS 1, November 2002.

[sha-1] National Institute of Standards and Technology, "Secure Hash Standard", FIBS PUB 180-1, April 1995, http://www.itl.nist.gov/fipspubs/fip180-1.htm.

Author's Addresses

Sheng Jiang
Huawei Technologies Co., Ltd
Huawei Building, No.3 Xinxi Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing
P.R. China
Email: jiangsheng@huawei.com

Sean Shen
CNNIC
4, South 4th Street, Zhongguancun
Beijing 100190
P.R. China
Email: shenshuo@cnnic.cn

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

W. Dec, Ed.
Cisco Systems
T. Mrugalski
Gdansk University of Technology
T. Sun
China Mobile
B. Sarikaya
Huawei USA
March 14, 2011

DHCPv6 Route Option
draft-ietf-mif-dhcpv6-route-option-01

Abstract

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

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on September 15, 2011.

Copyright Notice

Table of Contents

1. Introduction

The Neighbor Discovery (ND) ICMPv6 protocol [RFC4861] provides a mechanism for hosts to discover one or more default routers on a directly connected network segment. Extensions to the Router Advertisement (RA) protocol defined in [RFC4191] allow hosts to discover the preferences for multiple default routers on a given link, as well as any specific routes advertised by these routers. This allows network administrators to better handle multi-homed host topologies and influence the route selection by the host. This ND based mechanism however is sub optimal or impractical in some multi-homing scenarios, where DHCPv6 [RFC3315]is seen to be more viable.

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

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

2. Problem overview

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

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

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

3. DHCPv6 Based Solution

A DHCPv6 based solution allows an operator an on demand and node specific means of configuring static routing information. Such a solution also fits into network environments where the operator prefers to manage RG configuration information from a centralized DHCP server. [I-D.troan-multihoming-without-nat66] provides additional background to the need for a DHCPv6 solution to the problem.

In terms of the high level operation of the solution defined in this draft, a DHCPv6 client interested in obtaining routing information request the route option using the DHCPv6 Option Request Option (ORO) sent to a server. A Server, when configured to do so, provides the requested route information as part of a nested options structure covering; the next-hop address; the destination prefix; the route metric; any additional options applicable to the destination or next-hop. The overall DHCPv6 design follow a similar approach to that used in the design of the IA_NA, IA_TA and IA_PD options in [RFC3633]

4. DHCPv6 Route Option

A DHCPv6 client interested in obtaining routing information includes the OPTION_IA_RT as par of its DHCPv6 Option Request Option (ORO) in messages directed to a server (as allowed by [RFC3315], ie Solicit, Request, Renew, Rebind, Confirm or Information-request messages). A Server, when configured to do so, provides the requested route information using the OPTION_IA_RT option in messages sent in response (Advertise, and Reply). So as to allow the route option to be both extensible, as well as conveying detailed info for routes, use is made of a nested options structure. An IA_RT conveys one or

more OPTION_NEXT_HOP options that specify the IPv6 next hop addresses. Each OPTION_NEXT_HOP conveys in turn one or more OPTION_RT_PREFIX options that represents the IPv6 destination prefixes reachable via the given next hop. The Formats of the OPTION_IA_RT, OPTION_NEXT_HOP and OPTION_RT_PREFIX are defined in the following sub-sections

The DHCPv6 Route Option format borrows from the principles of the Route Information Option defined in [RFC4191]. One notable exception with respect to [RFC4191] is however that a Route Lifetime element is not defined. The information conveyed by the DHCPv6 Route Option is considered valid until changed or refreshed by general events that trigger DHCPv6 or route table state changes on a node, thus not requiring a specific route lifetime. In the event that it is desired for the client to request a refresh of the route information (and other stateless DHCPv6 options), use of the generic DHCPv6 Information Refresh Time Option, as specified in [RFC4242] is envisaged.

4.1. DHCPv6 Route Option Format

To separate routing information from other options conveyed in a DHCPv6 message, the DHCPv6 Route Option is defined and is used to convey to a client one or more IPv6 routes. Each IPv6 route consists of an IPv6 next hop address, an IPv6 destination prefix (a.k.a. the destination subnet), and a host preference value for the route. Elements of such route (e.g. Next hops and prefixes associated with them) are conveyed in IA_RT's options, rather than in the IA_RT option itself.

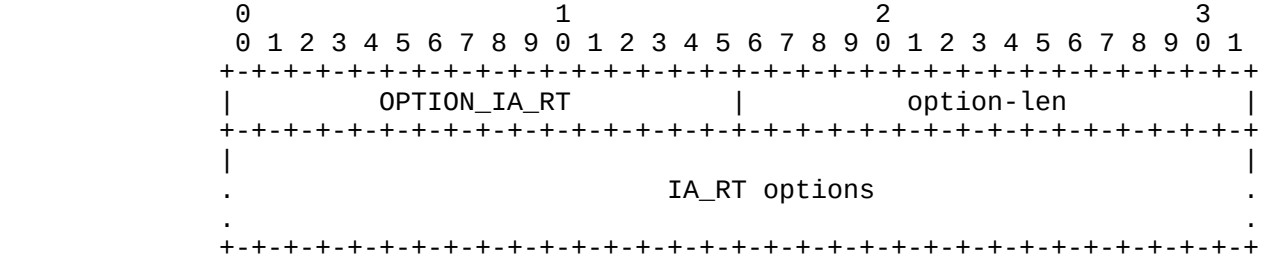


Figure 1: IPv6 Routes Option Format

option-code: OPTION_IA_RT (TBD).

option-len: Length of the IA_RT options field.

IA_RT options: Options associated with this IA_RT. This includes, but is not limited to, OPTION_NEXT_HOP options that specify next hop addresses.

The Route option MUST NOT appear in the following DHCPv6 messages: Solicit, Request, Renew, Rebind, Information-Request. The Route Option MAY appear in ADVERTISE and REPLY messages.

Discussion: Traditionally, grouping options (IA_NA, IA_TA and IA_RD) contain an identifier field (IAID) that must be unique among identifiers generated by one client. It is used to differentiate between several options of the same type (e.g. several IA_NA options) that may be used simultaneously. However, it is assumed that client will never use more than one IA_RT option therefore such an identifier is not needed.

4.2. Next Hop Option Format

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

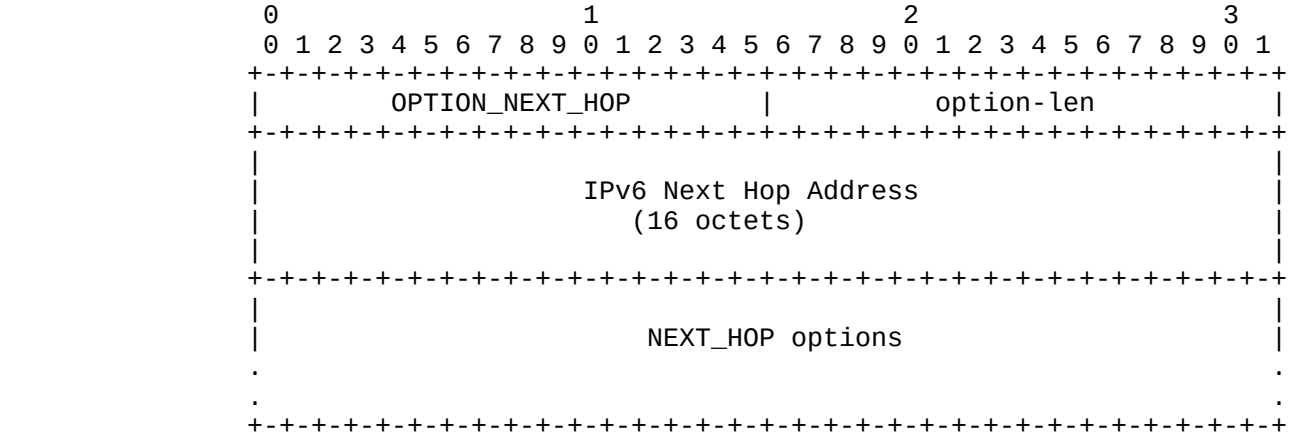


Figure 2: IPv6 Route Option Format

option-code: OPTION_NEXT_HOP (TBD).

option-len: 16 + Length of NEXT_HOP options field.

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

NEXT_HOP options: Options associated with this Next Hop. This includes, but is not limited to, one or more OPTION_RT_PREFIX options that specify prefixes reachable through the given next hop.

4.3. Route Prefix Option Format

The Route Prefix Option is used to convey information about a single prefix that represents the destination network. The Route Prefix Option is used as a sub-option in the previously defined Next Hop Option.

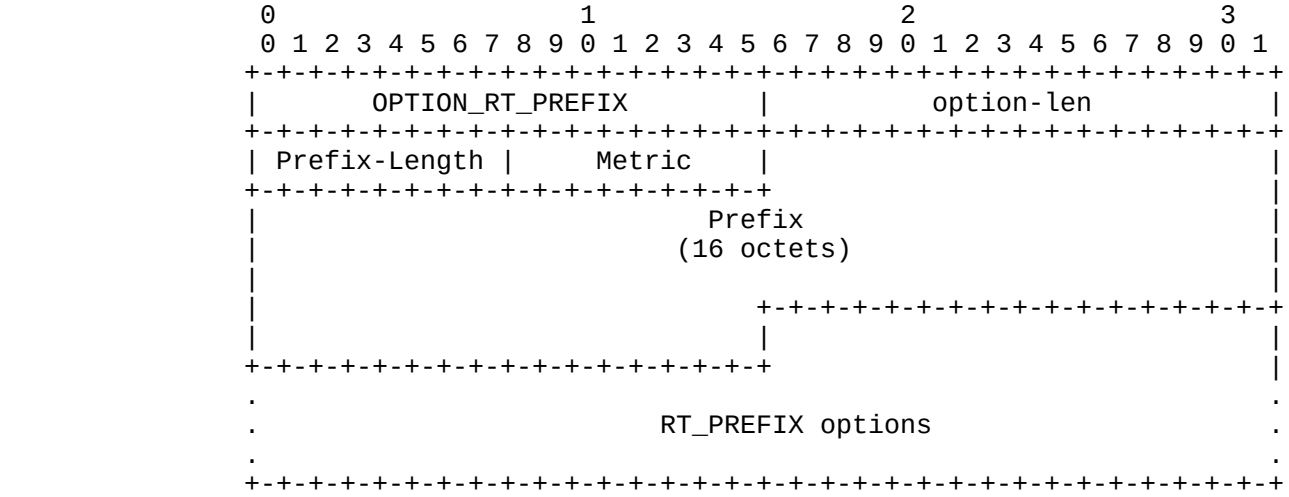


Figure 3: Route Prefix Option Format

option-code: OPTION_RT_PREFIX (TBD).

option-len: 18 + length of RT_PREFIX options.

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

Metric: Route Metric. 8-bit signed integer. The Route Metric indicates whether to prefer the next hop associated with this prefix over others, when multiple identical prefixes (for different next hops) have been received.

Prefix: Fixed length 16 octet field containing an IPv6 prefix.

RT_PREFIX options: Options specific to this particular prefix.

5. DHCPv6 Server Behavior

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

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

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

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

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

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

Discussion: How should server indicate that there are no specific routes for this particular client? The reasonable behavior is to return empty IA_RT option, possibly with Status Code indicating Success. Another approach could be to simply not return any IA_RT option.

6. DHCPv6 Client Behavior

A DHCPv6 client compliant with this specification MUST request the Route Option (option value TBD) in an Option Request Option (ORO) in the following messages: Solicit, Request, Renew, Rebind, Information-Request or Reconfigure. The messages are to be sent as and when specified by [RFC3315].

When processing a received Route Option a client MUST substitute a received 0::0 value in the Next Hop Option with the source IPv6 address of the received DHCPv6 message. It MUST also associate a received Link Local next hop addresses with the interface on which the client received the DHCPv6 message containing the route option. Such a substitution and/or association is useful in cases where the DHCPv6 server operator does not directly know the IPv6 next-hop

address, other than knowing it is that of a DHCPv6 relay agent on the client LAN segment. DHCPv6 Packets relayed to the client are sourced by the relay using this relay's IPv6 address, which could be a link local address.

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

The routes conveyed by the Route Option should be considered as complimentary to any other static route learning and maintenance mechanism used by, or on the client with one modification: The client MUST flush DHCPv6 installed routes following a link flap event on the DHCPv6 client interface over which the routes were installed. This requirement is necessary to automate the flushing of routes for clients that may move to a different network.

7. IANA Considerations

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

8. Security Considerations

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

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

9. Contributors and .Acknowledgements

This document would not have been possible without the significant contribution provided by: Arifumi Matsumoto, Hui Deng, Richard Johnson, Zhen Cao.

The authors would also like to thank Alfred Hines, Ralph Droms, Ted Lemon, Ole Troan, Dave Oran and Dave Ward for their comments and useful suggestions.

10. References

10.1. Normative References

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

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

10.2. Informative References

[I-D.troan-multihoming-without-nat66] Troan, O., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", draft-troan-multihoming-without-nat66-01 (work in progress), July 2010.

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

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

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

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

Authors' Addresses

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

Email: wdec@cisco.com

Tomasz Mrugalski
Gdansk University of Technology
Storczykowa 22B/12
Gdansk 80-177
Poland

Phone: +48 698 088 272
Email: tomasz.mrugalski@eti.pg.gda.pl

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

Phone:
Email: suntao@chinamobile.com

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

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

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: September 7, 2011

S. Jiang
Huawei Technologies Co., Ltd
G. Chen
China Mobile
March 4, 2011

Requirements for Addresses Registration
draft-jiang-6man-addr-registration-req-02.txt

Status of this Memo

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

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

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

This Internet-Draft will expire on September 7, 2011.

Copyright Notice

Abstract

In the IPv6 address allocation scenarios, node self-generated addresses are notionally conflicted with the network managed address architecture. These addresses need to be registered in the networking management plate for the purposes of central address administration. This document discusses the requirements of address registration and analyzes the possible solutions.

Table of Contents

1. Introduction & Requirements

In the IPv6 address allocation scenarios, node self-generated addresses, such as addresses in IPv6 Stateless Address Configuration [RFC4862, RFC4941] scenario and Cryptographically Generated Addresses (CGA, [RFC3972]), is notionally conflicted with the network managed address architecture, such as DHCPv6-managed network or network with Access Control List, in which addresses are assigned and managed by the network management plate.

The current IPv4 address allocation mode in DHCPv4-managed network is that the DHCPv4 server assigns addresses. Many operators of enterprise networks and similarly tightly administered networks have expressed the desire to hold on to this model when moving to IPv6, because they don’t want to have hosts end up with essentially random IPv6 addresses. However, the notion that a server assigns an address is for the most part incompatible with IPv6 stateless configuration.

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

This document discusses the requirements of address registration and analyzes the possible solutions. Dynamic Host Configuration Protocol for IPv6 (DHCPv6) and Router Advertisement may be extended to propagate the address registration request from network management to nodes. A DHCPv6 server may play the address registration server with newly defined DHCPv6 options. However, this may conflict with the original DHCP notion. A new set of protocol may have to be defined for the address registration purpose.

2. Terminology

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

3. Potential Solutions

3.1. Generic Address Registration Procedure

By current default, the nodes with self-generated addresses do not register their addresses to any network devices. However, this may result that the network may reject the access request from these devices.

As showed in below Figure 1, in the generic address registration procedure, the network management plate should firstly propagate the request of registering self-generated addresses. By received such requests, a node using the self-generated address should send an address registration message to the network management. The network management should check whether the requested address is accepted, for example, performing a Duplicated Address Detect or checking the address does not use the Reserved IPv6 Interface Identifiers [RFC5453]. If the requested address is accepted by the network management, it is registered in the address manage database, which may be used by other network functions, such as DNS or ACL. An acknowledgement is sent to the node, granting the usage of this address. If the requested address is not accepted by the network management, a rejected acknowledgement is sent to the node to indicate that it must generate a new address.

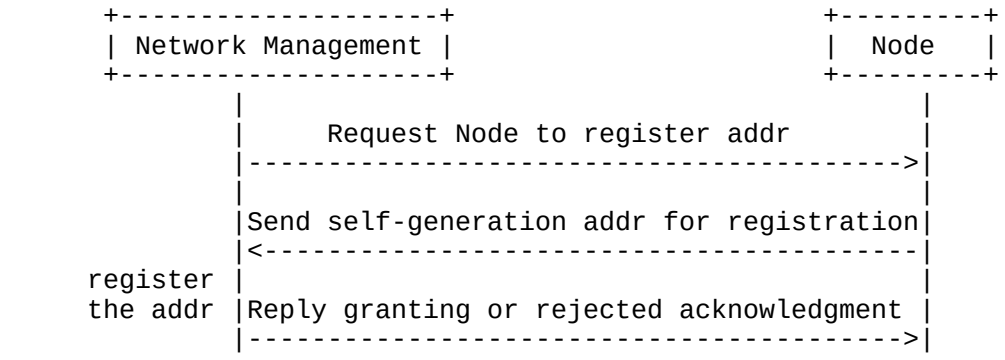


Figure 1: address registration procedure

3.2. Propagating the Registration Request

In order to indicate or force the nodes with self-generated addresses to register their addresses and the appointed address registration server, a new request option needs to be defined.

There are more than one mechanisms in which configuration parameters could be pushed to the end hosts. The address registration request option can be carried in Router Advertisement. In the DHCPv6 managed network, it can also be carried in DHCPv6 messages.

By receiving attendant of the address registration request option, a node MUST register its self-generated addresses, if there are any, to the appointed registration server. The option may be defined to include the default/enforced address registration server.

3.3. Address Registration Server and Protocol

In order to manage the address, an address registration server is needed with the support a set of address registration protocol.

The server should hold all registered addresses. It also needs to check whether the addresses meet the network address management policy, also performing a Duplicated Address Detect or checking the address does not use the Reserved IPv6 Interface Identifiers [RFC5453], etc. Its address data may be used by other network functions, such as DNS or ACL.

A set of address registration protocol need to at least support a basic information exchange: the node sends its address to the server and an acknowledgement is sent to the node.

3.3.1. Using DHCPv6 and DHCPv6 server

The current DHCPv6 protocol can be reused as the address registration protocol while a DHCPv6 serve plays as address registration server.

The current DHCPv6 specification allows for a host to communicate a set of "preferred" addresses to the server by listing these addresses in IA options [RFC3315]. In order to response to registration requests, an acknowledgement DHCPv6 option should be defined. It is used to indicate whether the registration of an IPv6 address is accepted.

3.3.2. Defining a new address Registration Protocol

However, the address registration procedure using DHC protocol may conflict with the initial notional of DHC protocol. The DHC protocol was originally designed to push configuration information from the network management side to the hosts while the address registration procedure is collecting information from hosts to the network management side.

A new set of address registration protocol may be defined.

[Author notes for IETF discussion:] Any other existing protocol may be used for address registration purposes?

4. Security Considerations

An attacker may use a faked address registration request option to indicate hosts reports their address to a malicious server and collect the user information. These attacks may be prevented by using secure protocols, in Neighbor Discovery protocol case, Secure Neighbor Discovery (SEND, [RFC3971]); in DHCP case, Secure DHCP [I-D.ietf-dhc-secure-dhcpv6]; or other additional security mechanisms.

An attacker could generate IPv6 address registration requests in order to exhaust the server resources (or to impact on any other operation that depend on the registration of the address).

In the use case of DHCPv6, the address registration procedure is as vulnerable as all other mechanisms based on DHCPv6 to DOS attacks to the server. Proper use of DHCPv6 autoconfiguration facilities [RFC3315], such as AUTH option or Secure DHCP [I-D.jiang-dhc-secure-dhcpv6] can prevent these threats.

5. IANA Considerations

There is no IANA considerations.

6. Change Log [RFC Editor please remove]

draft-jiang-6man-addr-registration-req-00, original version, 2010-03-01

draft-jiang-6man-addr-registration-req-01, minor update, 2010-08-27

draft-jiang-6man-addr-registration-req-02, minor update, 2010-03-04

7. Acknowledgments

The authors would like to thank Cao Wei, Huawei for been involved in the early requirement identification and early discussion.

8. References

8.1. Normative References

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

[RFC3315] R. Droms, Ed., J. Bound, B. Volz, T. Lemon, C. Perkins and M. Carne, "Dynamic Host Configure Protocol for IPv6", RFC3315, July 2003.

[RFC3971] J. Arkko, J. Kempf, B. Zill, P. Nikander, "SEcure Neighbor Discovery (SEND) ", RFC 3971, March 2005.

[RFC3972] T. Aura, "Cryptographically Generated Address", RFC3972, March 2005.

[RFC4862] S. Thomson, T. Narten and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC4862, September 2007.

[RFC4941] T. Narten, R. Draves and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC5453] S. Krishnan, "Reserved IPv6 Interface Identifiers", RFC 4543, February 2009.

8.2. Informative References

[I-D.ietf-dhc-secure-dhcpv6] S. Jiang and S. Shen "Secure DHCPv6 Using CGAs", draft-ietf-dhc-secure-dhcpv6-02 (work in progress), December, 2010.

Author's Addresses

Sheng Jiang
Huawei Technologies Co., Ltd
Huawei Building, No.3 Xinxi Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing 100085
P.R. China
Email: jiangsheng@huawei.com

Gang Chen
China Mobile
53A,Xibianmennei Ave.,
Xuanwu District,
Beijing 100053
China
Email: phdgang@gmail.com

DHC
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2012

T. Mrugalski
ISC

July 02, 2011

DHCPv6 Options for IPv4 Residual Deployment (4rd)
draft-mrugalski-dhc-dhcpv6-4rd-00

Abstract

This document specifies DHCPv6 options which are meant to be used by 4rd Customer Edge (CE) to obtain necessary parameters to configure their 4rd automatic tunnels. The 4rd architecture is defined in [I-D.despres-intarea-4rd]. Since specification of 4rd is still expected to evolve, DHCPv6 options may have to evolve too to fit the revised 4rd specification.

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 3, 2012.

Copyright Notice

Table of Contents

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

IPv4 Residual Deployment across IPv6-Service networks (4rd) [I-D.despres-intarea-4rd] is an automatic tunneling mechanism for providing IPv4 connectivity service to end users over a service provider's IPv6 network. To operate properly, 4rd Customer Edge (CE) requires one or more mapping rules configured. One mapping rule constists of the following parameters: Length of CE IPv6 Prefix, Domain IPv6 Prefix, Domain 4rd Prefix, and optionally Domain IPv6 suffix. Also, Anycast BR IPv6 address must be configured for proper operation. This document defines several DHCPv6 options that allow delivery of required information to configure CE. Definitions of enumerated parameters are provided in [I-D.despres-intarea-4rd]. Since specification of 4rd is still expected to evolve, DHCPv6 options may have to evolve too to fit the revised 4rd specification.

3. DHCPv6 Options Format

To configure CE equipment remotely, DHCPv6 should be used. Several new options are defined for that purpose. Their format and usage is defined in the following sections.

Discussion: Proposed layout assumes that several simple options are used. Such approach simplifies implementation as it is much easier for implementors to reuse existing code handling such options. This design choice comes at a cost, however. Clients must perform checks if provided set of options is complete. Alternatively, it would be possible to define one complex option that contains all mandatory parameters. Nevertheless, since postfix parameter is optional, it would still require sub-options (or conditional formatting that is strongly not recommended [I-D.ietf-dhc-option-guidelines]).

Discussion: It has also been proposed that since 3 mandatory parameters - Domain IPv6 Prefix, Length of the CE IPv6 Prefix (note: these are different prefixes), and Domain 4rd prefix - are always present, they may be grouped together.

3.1. 4rd Options Cardinality

To properly configure 4rd infrastructure, following parameters are required:

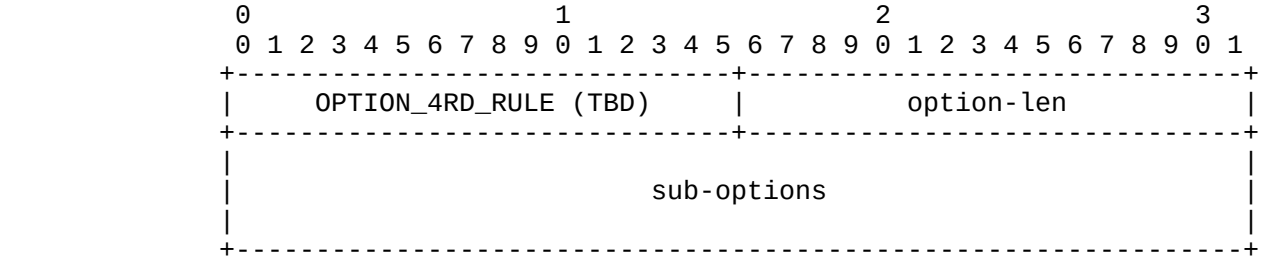
Exactly one Anycast BR IPv6 Address.

One or more 4rd mapping rules. Each 4rd mapping rule consists of exactly one 4rd CE prefix, exactly one CE IPv6 prefix length, and exactly one 4rd prefix. Each mapping rule may contain zero or one Domain IPv6 Suffix.

3.2. 4rd Mapping Rule Option

The 4rd Mapping Rule Option does not convey any information on its own, but rather is used to group options that represent parameteres of a single mapping rule. 4rd architecture allows more than one mapping rules, therefore the 4rd Mapping Rule Option MAY appear more than once in a DHCPv6 message.

The format of the 4rd Mapping Rule Option is shown in Figure 1.



option-code: OPTION_4RD_RULE (TBD)

option-len: Length of all sub-options.

sub-options: options defining Mapping Rule parameters

Figure 1: 4rd Mapping Rule Option

The 4rd Mapping Rule Option consists of option-code and option-len fields (as all DHCPv6 options do), and a variable length sub-options field that contains rule parameters.

The 4rd Mapping Rule Option SHOULD NOT appear in any other than the following DHCPv6 messages: Solicit, Advertise, Request, Renew, Rebind, Information-Request and Reply. The 4rd Rule Option may appear more than once in each message.

Each 4rd Mapping Rule Option MUST contain exactly one 4rd CE Prefix Option, exactly one CE IPv6 Prefix Length Option, exactly one 4rd Prefix Option and zero or one Domain IPv6 Suffix Option. Option that does not meet this criteria is invalid and MUST be ignored.

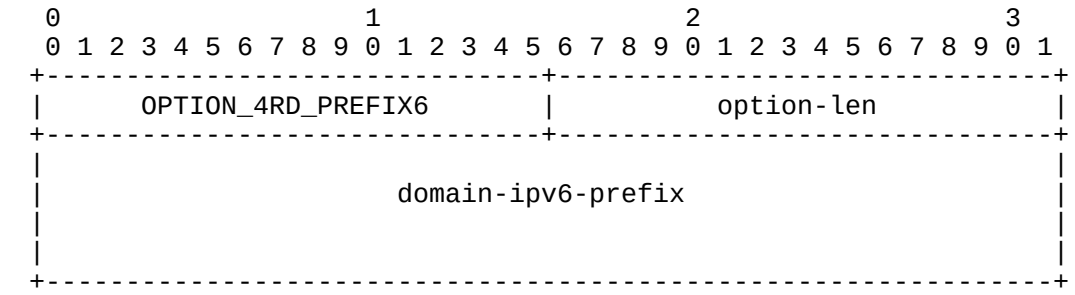
4rd Mapping Rule Option may contain additional options.

Discussion: Defined format does not allow referencing rules, i.e. if there are several rules, there is no rule 1, rule 2 etc. DHCPv6 protocol does not allow leveraging order in which options appear in the message. If rule indentification is useful, a small ID field may be added to the 4rd Mapping Rule Option.

3.3. CE IPv6 Prefix Option

CE IPv6 Prefix Option conveys information about domain IPv6 prefix that is going to be used by requesting client (CE).

The format of the CE IPv6 Prefix Option is defined in Figure 2.



option-code: OPTION_4RD_PREFIX6 (TBD)

option-len: Length of Domain IPv6 Prefix in octets (16)

domain-ipv6-prefix: 4rd Domain IPv6 Prefix

Figure 2: CE IPv6 Prefix Option

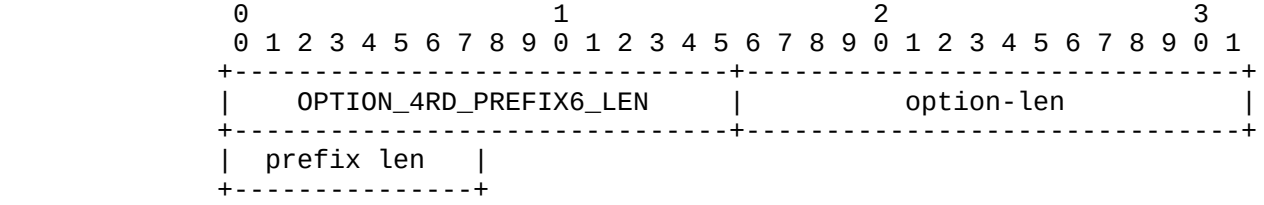
The CE IPv6 Prefix Option consists of option-code and option-len fields (as all DHCPv6 options do), and 16 octets long Domain IPv6 Prefix field.

The CE IPv6 Prefix Option MUST NOT appear in DHCPv6 message directly. It MUST NOT appear in any option other than 4rd Mapping Rule Option.

3.4. CE IPv6 Prefix Length Option

The CE IPv6 Prefix Length Option defines length of the prefix assigned to specific CE. It is expected to be used together with CE IPv6 Prefix Option, defined in Section Section 3.3.

The format of the CE Prefix Length Option is shown in Figure 3.



option-code: OPTION_4RD_PREFIX6_LEN: (TBD)

option-len: Length of the prefix-len field in octets (1)

prefix-len: Length of Domain IPv6 Prefix

Figure 3: CE IPv6 Prefix Length Option

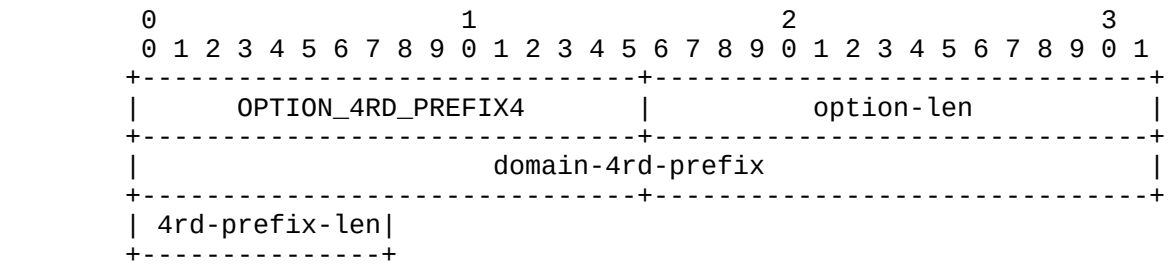
The CE IPv6 Prefix Length Option consists of option-code and option-len fields (as all DHCPv6 options do), and 1 octet long Domain IPv6 Prefix Length field that specifies length in bits (0-64).

The CE IPv6 Prefix Length Option MUST NOT appear in DHCPv6 message directly. It MUST NOT appear in any option other than 4rd Mapping Rule Option.

3.5. Domain 4rd Prefix Option

The Domain 4rd Prefix Option contains IPv4 address. Depending on its length it is IPv4 prefix, an IPv4 address, or a shared IPv4 address followed by Port-set ID.

The format of the Domain 4rd Prefix Option is shown in Figure Figure 4.



OPTION_4RD_PREFIX: (TBD)

option-len: 7

domain-4rd-prefix: Domain 4rd Prefix (an IPv4 address)

4rd-prefix-len: Length of Domain IPv6 Prefix (in bits)

Figure 4: Domain 4rd Prefix Option

The Domain 4rd Prefix Option consists of option-code and option-len fields (as all DHCPv6 options do), four octets long domain-4rd-prefix field that contains IPv4 address and one octect long 4rd-prefix-len.

Depending on 4rd-prefix-len value, actual 4rd Prefix may be range of IPv4 addresses (part of domain-4rd-prefix), a single IPv4 address (specified by domain-4rd-prefix) or IPv4 address+port range (specified by domain-4rd-prefix and Port-set ID). Port-Set ID is derived from CE Index, as explained in see Section 4.3.3 in [I-D.despres-intarea-4rd].

3.6. Domain IPv6 Suffix Option

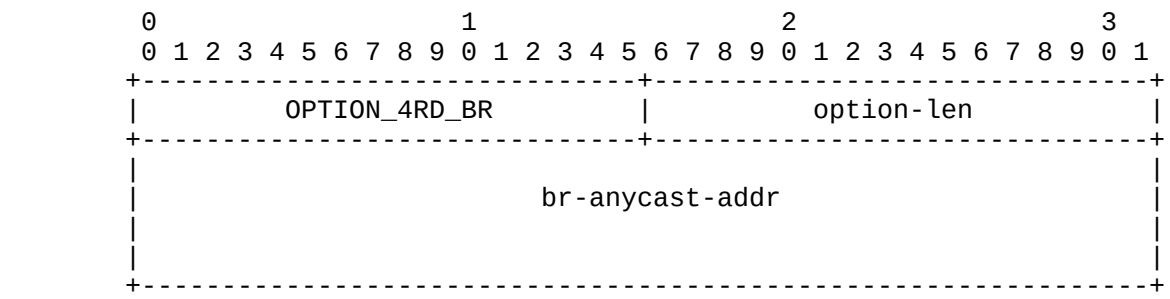
TODO: I don't understand what this suffix is. [I-D.despres-intarea-4rd] is unclear. My understanding is that suffix are bits that are appended to Domain IPv6 Prefix + CE Index and they together form CE IPv6 Prefix. It is only used when Domain IPv6 Prefix and CE Index are short enough. On the other hand, slide 4 of http://tools.ietf.org/agenda/80/slides/dhc-6.pdf suggests that suffix defines something that is appended after CE IPv6 Prefix.

Discussion: Wouldn't it be better to just use Domain IPv6 Prefix as a template with CE Index inserted at appropriate offset? This would make configuration simpler (one less option) and also validation of received option would be much more straightforward.

3.7. BR Anycast Option

The BR Anycast Option specifies an IPv6 anycast address that should be used to reach nearest Border Relay.

The format of the BR Anycast Option is shown in Figure 5.



option-code: OPTION_4RD_BR (TBD)

option-len: Length of BR IPv6 Anycast Prefix (16)

br-anycast-addr: Border Relay IPv6 Anycast Address

Figure 5: BR IPv6 Address Option

The BR IPv6 Address Option consists of option-code and option-len fields (as all DHCPv6 options do), and a 16 octets long br-anycast-addr field that specifies Border Relay IPv6 Anycast address.

The BR Anycast Option SHOULD NOT appear in any other than the following DHCPv6 messages: Solicit, Advertise, Request, Renew, Rebind, Information-Request and Reply.

The BR Anycast Option MAY appear zero or one time in a single message.

4. 4rd Options Example

TODO: Provide an example here. Possibly reuse example from Remi’s presentation from IETF80.

5. DHCPv6 Server Behavior

Server conformant to this specification MUST allow configuration of one or more Mapping Rule Options.

Server MUST transmist all configured instances of the Mapping Rule Options with all sub-options, if client requested it using OPTION_4RD_RULE in ORO.

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 4rd Mapping Rule Option if the client has not explicitly enumerated it on its Option Request Option.

6. DHCPv6 Client Behavior

Although other use cases are allowed, in typical use case CE will act as DHCPv6 client and will request 4rd configuration to be assigned by the DHCPv6 server located in the ISP network. A client that supports 4rd CE functionality and conforms to this specfication MUST include OPTION_4RD_RULE in its ORO.

Client SHOULD request 4rd options in SOLICIT, REQUEST, RENEW, REBIND and INFORMATION-REQUEST messages.

If the client receives OPTION_4RD_RULE option, it must verify the option contents, as described in Section 3.2. In case of failved verification, client MUST discard invalid option and continue processing any following options, including other instances of 4rd options.

If client receives more than one OPTION_4RD_RULE, it MUST use all received instances. It MUST NOT use only the first one, while discarding remaining ones.

Note that system implementing 4rd CE functionality may have multiple network interfaces, and these interfaces may be configured differently; some may be connected to networks that call for 4rd, and some may be connected to networks that are using normal dual stack or other means. The 4rd CE system should approach this specification on an interface-by-interface basis. For example, if the CE system is attached to multiple networks that provide the 4rd Mapping Rule Option, then the CE system MUST configure a 4rd tunnel for each interface separately as each 4rd provides IPv4 connectivity for each distinct interface. Means to bind a 4rd configuration to a given interface in a multiple interfaces device are out of scope of this document.

7. Security Considerations

Implementation of 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 4rd can be trusted, and it should not therefore bypass any security mechanisms such as IP firewalls.

Section 23 of [RFC3315] discusses DHCPv6-related security issues.

Section 6 of [I-D.despres-intarea-4rd] discusses 4rd related security issues.

8. IANA Considerations

IANA is requested to allocate DHCPv6 option codes referencing this document, delineating OPTION_4RD_RULE, OPTION_4RD_PREFIX6, OPTION_4RD_PREFIX6_LEN, OPTION_4RD_PREFIX and OPTION_4RD_BR option names.

9. Acknowledgements

This document would not have been possible without support from Remi Despres, Satoru Matsushima, Ole Troan and Tetsuya Murakami.

10. References

10.1. Normative References

[I-D.despres-intarea-4rd]
Despres, R., Matsushima, S., Murakami, T., and O. Troan, "IPv4 Residual Deployment across IPv6-Service networks (4rd) ISP-NAT’s made optional", draft-despres-intarea-4rd-01 (work in progress), March 2011.

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

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

10.2. Informative References

[I-D.ietf-dhc-option-guidelines]
Hankins, D., "Guidelines for Creating New DHCP Options", draft-ietf-dhc-option-guidelines-06 (work in progress), March 2010.

Author's Address

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

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

Dynamic Host Configuration (DHC)
Internet-Draft
Intended status: Informational
Expires: December 28, 2011

T. Mrugalski, Ed.
ISC
K. Kinnear
Cisco

June 26, 2011

DHCPv6 Failover Requirements
draft-mrugalski-dhc-dhcpv6-failover-requirements-00

Abstract

The DHCPv6 protocol, defined in [RFC3315] allows for multiple servers to operate on a single network, however it does not define any way to decide which server responds to which client queries. Some sites are interested in running multiple servers in such a way as to provide increased availability in case of server failure. In order for this to work reliably, the cooperating primary and secondary servers must maintain a consistent database of the lease information. [RFC3315] allows for but does not define any redundancy or failover mechanisms. This document outlines requirements for DHCPv6 failover, enumerates related problems, and discusses the proposed scope of work to be conducted. This document does not define a DHCPv6 failover protocol.

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 December 28, 2011.

Table of Contents

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

The DHCPv6 protocol, defined in [RFC3315] allows for multiple servers to be operating on a single network, however it does not define any way to decide which server responds to which client queries. Some sites are interested in running multiple servers in such a way as to provide redundancy in case of server failure. In order for this to work reliably, the cooperating primary and secondary servers must maintain a consistent database of the lease information.

This document discusses failover implementations scenarios, failure modes, and synchronization approaches to provide background to the list of requirements for a DHCPv6 failover protocol. It then defines a minimum set of requirements that failover must provide to be useful, while acknowledging that additional features may be specified as extensions. This document does not define a DHCPv6 failover protocol.

3. Definitions

This section defines terms that are relevant to DHCPv6 failover.

Definitions from [RFC3315] are included by reference. In particular, client means any device (e.g., end user host, CPE or other router) that implements client functionality of the DHCPv6 protocol. A server means a DHCPv6 server, unless explicitly noted otherwise. A relay is a DHCPv6 relay.

A binding (or, client binding) is a group of server data records containing the information the server has about the addresses in an IA or configuration information explicitly assigned to the client. Configuration information that has been returned to a client through a policy - for example, the information returned to all clients on the same link - does not require a binding.

DDNS - an abbreviation for "Dynamic DNS", which refers to the capability to update a DNS server’s name database using the on-the-wire protocol defined in [RFC2136]. Clients and servers can negotiate the scope of such updates as defined in [RFC4704].

Failover - an ability of one partner to continue offering services provided by another partner, with minimal or no impact on clients.

FQDN - a fully qualified domain name. A fully qualified domain name generally is a host name with at least one zone name. For example "dhcp.example.org" is a fully qualified domain name.

High Availability - a desired property of DHCPv6 servers to continue providing services despite experiencing unwanted events such as server crashes, link failures, or network partitions.

Load Balancing - the ability for two or more servers to each process some portion of the client request traffic in a conflict-free fashion.

Lease - an IPv6 address, an IPv6 prefix or other resource that was assigned ('leased') by a server to a specific client. A lease may include additional information, like associated fully qualified domain name (FQDN) and/or information about associated DNS updates.

Partner - A "partner", for the purpose of this document, refers to a failover server, typically the other failover server in a failover relationship.

Stable Storage - each DHCP server is required to keep its lease database in some form of storage (known as "stable storage") that will be consistent throughout reboots, crashes and power failures.

Partner Failure - A power outage, unexpected shutdown, crash or other type of failure that renders a partner unable to continue its operation.

4. Scope of work

In order to fit within the IETF process effectively and efficiently, the standardization effort for DHCPv6 failover is expected to proceed with the creation of documents of increasing specificity. It begins with this document specifying the requirements for DHCPv6 failover ("requirements document"). Later documents are expected to address the design of the DHCPv6 failover protocol ("design document"), and if sufficient interest exists, the protocol details required to implement the DHCPv6 failover protocol itself ("protocol document"). The goal of this partitioning is, in part, to ease the validation, review, and approval of the DHCPv6 failover protocol by presenting it in comprehensible parts to the larger community.

Additional documents describing extensions may also be defined.

DHCPv6 Failover requirements are presented in Section 8.

4.1. Failover alternatives

There are many scenarios when it seems that a failover capability would be useful. However, there are often much simpler approaches that will meet the required goals. This section documents examples where failover is not really needed.

4.1.1. Short-lived addresses

There are cases when IPv6 addresses are used only for a short time, but there is a need to have high degree of confidence that those addresses will be served. A notable example is a scenario where hosts require an address during boot. Address and possibly other configuration parameters are used during boot process and are discarded afterwards. Any lack of available DHCPv6 service at this time may render the devices unbootable.

Instead of deploying failover, it is better to use the much simpler preference mechanism, defined in [RFC3315]. For example, consider two or more servers with each having distinct preference set (e.g. 10 and 20). Both will answer to a client's request. The client should choose the one with larger preference value. In case of failure of the most preferred server, the next server will keep responding to clients' queries.

5. Failover Scenarios

The following section provides several examples of deployment scenarios and use cases that may be associated with capabilities commonly referred to as failover. These scenarios may be inside or outside of scope for DHCPv6 failover protocol as defined by this document. They are enumerated here to provide a common basis for discussion.

5.1. Hot Standby Model

In the simplest case, there are two partners that are connected to the same network. Only one of partners ('primary') provides services to clients. In case of its failure, the second partner ('secondary') continues handling services previously handled by first partner. As both servers are connected to the same network, a partner that fails to communicate with its partner while also receiving requests from clients may assume with high probability that its partner is down and

the network is functional. This assumption may affect its operation.

5.2. Geographically Distributed Failover

Servers may be physically located in separate locations. A common example of such a topology is where a service provider has at least a regional high performance network between geographically distributed datacenters. In such a scenario, one server is located in one datacenter and its failover partner is located in another remote datacenter. In this scenario, when one partner finds that it cannot communicate with the other partner, it does not necessarily mean that the other partner is down.

5.3. Load balancing

A desire to have more than one server in a network may also be created by the desire to have incoming traffic be handled by several servers. This decreases the load each server must endure when all servers are operational. Although such a capability does not, strictly, require failover - it is clear that failover makes such an architecture more straightforward.

Note that in a load balancing situation which includes failover, each individual server MUST be able to handle the full load normally handled by both servers working together, or there is not a true increase in availability.

5.4. 1-to-1, m-to-1 and m-to-m models

A failover relationship for a specific network is provided by two failover partners. Those partners communicate with each other. This scenario is sometimes referred to as the 1-to-1 model and is considered relatively simple. In larger networks one server may be participating in several failover relationships, i.e. it provides failover for several address or prefix pools, each served by separate partners. Such a scenario can be referred to as m-to-1. The most complex scenario - m-to-m - assumes that each partner participates in multiple failover relationships.

5.5. Split prefixes

Due to the extensive IPv6 address space, it is possible to provide semi-redundant service by splitting the available pool of addressees into two or more non-overlapping pools, with each server handling its own smaller pool. Several versions of such a scenario are discussed in [I-D.ietf-dhc-dhcpv6-redundancy-consider].

5.6. Long lived connections

Certain nodes may maintain long lived connections. Since the IPv6 address space is large, techniques exist (e.g. [I-D.ietf-dhc-dhcpv6-redundancy-consider]) that use the easy availability of IPv6 addresses in order to provide increased DHCPv6 availability. However, these approaches do not generally provide for stable IPv6 addresses for DHCPv6 clients should the server with which the client is interacting become unavailable.

6. Principles of DHCPv6 Failover

This section describes important issues that will affect any DHCPv6 failover protocol. This section is not intended to define implementation details, but rather high level concepts and issues that are important to DHCPv6 failover. These issues form a basis for later documents which deal with the solutions to these issues.

6.1. Failure modes

This section documents failure modes. Each failure mode is listed as either an in-scope or out-of-scope requirement for the failover protocol.

6.1.1. Server Failure

Servers may become unresponsive due to a software crash, hardware failure, power outage or any number of other reasons. The failover partner will detect such event due to lack of responses from the down partner. In this failure mode, the assumption is that the server is the only equipment that is off-line and all other network equipment is operating normally. In particular, communication between other nodes is not interrupted.

When working under the assumption that this is the only type of failure that can happen, the server may safely assume that its partner unreachability means that it is down, so other nodes (clients in particular) are not able to reach it either and no services are provided.

It should be noted that recovery after the failed server is brought back on-line is straightforward, due to the fact that it just needs to download current information from the lease database of the healthy partner and there is no conflict resolution required.

This is by far the most common failure mode between two failover partners.

When the two servers are located physically close to each other, possibly in the same room, the probability that a failure to communicate between failover partners is due to server failure is increased.

6.1.2. Network partition

Another possible cause of partner unreachability is a failure in the network that connects the two servers. This may be caused by failure of any kind of network equipment: router, switch, physical cables, or optic fibers. As a result of such a failure the network is split into two or more disjoint sections (partitions) that are not able to communicate with each other. Such an event is called network partition. If failover partners are located in different partitions, they won't be able to communicate with each other. Nevertheless, each partner may still be able to serve clients that happen to be part of the same partition.

If this failure mode is taken into consideration, a server can't assume that partner unreachability automatically means that its partner is down. They must consider the fact that the partner may continue operating and interacting with a subset of the clients. The only valid assumption is that partner also detected the network partition event and follows procedures specified for such a situation.

It should be noted that recovery after partitioned network is rejoined is significantly more complicated than recovery from a server failure event. As both servers may have kept serving clients, they have two separate lease databases, and they need to agree on the state of each lease (or follow any other algorithm to bring their lease databases into agreement).

This failure mode is more likely (though still rare) in the situation where two servers are in physically distant locations with multiple network elements between them. This is the case in geographically distributed failover (see Section 5.2).

6.2. Synchronization mechanisms

Partners must exchange information about changes made to the lease database. There are two types of sychronization methods that may be used.

6.2.1. Lockstep

When a server receives a REQUEST message from a client it consults its lease database and assigns requested addresses and/or prefixes.

To make sure that its partner maintains a consistent database, it then sends information about new or just updated lease to the partner and waits for the partner's response. After the response from partner is received the REPLY message is transmitted to the client.

This approach has the benefit of having a completely consistent lease database between partners at all times. Unfortunately, there is a large performance penalty for this approach as each response sent to a client is delayed by the total sum of the delays caused by two transmissions between partners and the processing by the second partner. The second partner is expected to update its own copy of the lease database in permanent storage, so this delay is not negligible, even in fast networks.

6.2.2. Lazy updates

Another approach to synchronizing the lease databases is to transmit the REPLY message to the client before completing the update to the partner. The server sends the REPLY to the client immediately after assigning appropriate addresses and/or prefixes and initiates the partner update at a later time, depending on the algorithm chosen. Another varation of this approach is to initiate transmission to the partner, but not wait for its response before sending the REPLY to the client.

This approach has benefit of a minimal impact on server response times, thus it is much better from a performance perspective. However, it makes the lease databases loosely synchronized between partners. This makes the synchronization more complex (and particularly the re-integration after a network partition event), as there may be cases where one client has been given a lease on an address or prefix of which the partner is not aware (e.g. if server crashes after sending REPLY to the client, but before sending update information to its partner).

7. DHCPv4 and DHCPv6 Failover Comparison

There are significant similarities between existing DHCPv4 and envisaged DHCPv6 failovers. In particular both serve IP addresses to clients, require maintaining consistent databases among partners, need to perform consistent DNS Updates, must be able take over bindings offered by failed partner, must be able to resume operation after partner is recovered. DNS conflict resolution works on the same principles in both DHCPv4 and DHCPv6.

Nevertheless, there are significant differences. IPv6 introduced prefix delegation [RFC3633] that is a crucial element of the DHCPv6

protocol. IPv6 also introduced the concept of deprecated addresses with separate prefered and valid lifetimes, both being configured via DHCPv6. Negative response (NACK) in DHCPv4 has been replaced with the ability in DHCPv6 to provide corrected response in a REPLY message that differs from a REQUEST.

Also, the typical large address space (close to 2^64 addresses on /64 prefixes expected to be available on most networks) may make managing address assignment significantly different from DHCPv4 failover. In DHCPv4 it was not possible to use a hash or calculated technique to divide the significantly more limited address space and therefore much of the protocol that deals with pool balancing and rebalancing might not be necessary and can be done mathematically. And, it may also be possible and reasonable to use a much longer MCLT value -- as reusing an address for a different client is generally not a requirement (at least over the near term) as close to 2^64 addresses may be available.

However, DHCPv6 Prefix Delegation is similar to IPv4 addressing and therefore techniques for pool balancing and rebalancing and shorter MCLT times will be needed.

8. DHCPv6 Failover Requirements

This section summarizes the requirements for DHCPv6 failover.

Certain capabilities may be useful in some, but not all scenarios. Such additional features will be considered optional parts of failover, and will be split and defined in separate documents. As such, this document can be considered an attempt to define requirements for the DHCPv6 failover 'core' protocol.

The core of the DHCPv6 failover protocol is expected to provide the following properties:

1. The number of supported partners MUST be exactly two, i.e. there are at most two servers that are aware of specific lease; this approach is often called 1-to-1 model.

2. For each prefix or address pool, server MUST NOT participate in more than one failover relationship.

3. Server MAY participate in multiple relationships only if those relationships cover different prefix or address pools.

4. One partner MUST be able to continue serving leases offered by the other partner. This property is also sometimes called

'lease stability'. The failure of either failover partner SHOULD pose minimal or no impact on client connectivity. In particular, it MUST NOT force the client to change addresses and/or change prefixes delegated to it. Long-lived connections MUST NOT be disturbed.

5. Prefix delegation MUST be supported.

6. Use of the failover protocol MUST NOT introduce significant performance impact on server response times. Therefore synchronization between partner MUST be done using some form of lazy updates (see Section 6.2.2).

7. The pair of failover servers MUST be able to recover from server down failure (see Section 6.1.1).

8. The pair of failover servers MUST be able to recover from a network partition event (see Section 6.1.2).

9. The design MUST allow secure communication.

10. The definition of extensions to this core protocol SHOULD be allowed, when possible.

High Availability is a property of the protocol that allows clients to receive DHCPv6 services despite the failure of individual DHCPv6 servers. In particular, it means the server that takes over providing service to clients from its failed partner, will continue serving the same addresses and/or prefixes. This property is also sometime called lease stability.

The core protocol MUST be limited to the 1-to-1 model (see Section 5.4). In addition, the core protocol MUST restrict each address or prefix pool to participate in at most one failover relationship. (Note: these are different statements!) If there is sufficient community support for failover servers to participate in more than one failover relationship (thus providing support for a form of m-to-1 deployment), this capability SHALL be defined as an extension to the core failover protocol.

Despite the lack of standardization of DHCPv4 failover, the coexistence of DHCPv4 and DHCPv6 failover MAY be taken into consideration. In particular, certain features that are common for both IPv4 and IPv6, like DNS Update mechanism SHOULD be taken into consideration.

A Load Balancing capability is considered an extension and MAY be defined in a separate document. It MUST NOT be part of the core

protocol, but rather defined as an extension. The primary reason for this the desire to limit core protocol complexity.

Following features and capabilities are outside of scope of this document:

m-to-m model (see section Section 5.4)

servers participating in multiple failover relationships

load balancing

9. Related work

This section describes related work. Readers may benefit from familiarizing themselves with these approaches, their advantages and limitations.

9.1. DHCPv4 failover concepts

9.1.1. Goals of DHCPv4 Failover

1. Provide a high availability DHCP service by leveraging the hooks built into DHCPv4 [RFC2131] and its usual implementation to support multiple servers able to respond to client requests. These hooks are:

   (a) The broadcast of DHCPDISCOVER requests.

   (b) The transition from unicast for DHCPREQUEST/RENEW to broadcast for DHCPREQUEST/REBIND.

   (c) The usual implementation of DHCPv4 relay agents to allow forwarding of DHCPv4 requests to multiple different DHCPv4 servers.

2. Produce a minimal impact on performance of the DHCPv4 server.

3. Prevent duplicate IP address allocation even in the event of a network partition.

4. Allow multiple failover relationships per server.

5. Standardize only the minimum necessary to provide a high availability DHCP service. In particular, avoid standardizing the interchange of configuration information.

9.1.2. Goals lead to Concepts

The goal to have a minimal performance impact on the operation of the DHCPv4 servers participating in failover is the driving force behind the design of the DHCPv4 failover protocol.

The steps in this chain of reasoning are as follows:

1. To avoid the major performance impact that a lockstep update of a failover partner would inflict, use a lazy update approach (see Section 6.2.2).

2. When using lazy update, there is always the problem that the failover server could crash after it has responded to some number of DHCPv4 clients and before it has updated its partner with the lease information it provided to those clients.

3. Thus, when one failover server cannot communicate with another failover server, it cannot know what that other failover server has told any of its DHCPv4 clients.

This creates an obvious problem.

The central concept in the DHCPv4 failover design is to place a limit on the uncertainty described in point #3, above. The DHCPv4 failover protocol is designed to ensure that every failover server knows at all times, not exactly what its failover partner has told to the DHCPv4 clients with which it is communicating, but rather the limits of what its failover partner could have told any DHCPv4 clients with which it was communicating.

This is done by ensuring that no DHCPv4 server participating in a failover relationship will ever offer a lease time to any DHCPv4 client that is more than an agreed-upon value beyond that known by its failover partner.

This agreed-upon value is called the "Maximum Client Lead Time", and abbreviated MCLT.

Thus, every DHCPv4 failover partner needs to know what its partner knows about every lease in the server, and it needs to ensure that it will never provide a lease time to any DHCPv4 client that is beyond what its partner believes the current lease time to be plus the MCLT.

Given this fundamental guarantee, if one failover server cannot communicate with its failover partner, then it knows the limits of what any DHCPv4 client of that missing partner might have for a lease time. If this failover server waits until it believes a lease has

expired and then also waits until the MCLT has passed, it knows that the lease is sure to have expired (or the DHCPv4 client will have tried to renew the lease and communicated with the remaining DHCPv4 server). (We will deal with network partition events below.)

In order to allow a remaining failover server to provide service to newly arrived DHCPv4 clients, while waiting out the MCLT beyond the lease expiration (if any), the protocol provides for allocation of some percentage of the available leases to each failover partner.

A failover server which cannot commuicate with its partner must therefore wait out the MCLT beyond the lease expiration (if any) of IP addresses before it can allocate them to DHCPv4 clients. This could impact the server's ability to provide available IP addresses to newly arrived DHCPv4 clients. To prevent this impact the DHCPv4 failover protocol divides the allocation of the available leases between each failover partner. The protocol supports periodic rebalancing of the allocation of these available leases.

9.1.3. Use of the MCLT in practice

From the above discussion, it should be clear how to avoid re-using an IP address before it has expired. The MCLT is central to the operation of the protocol. One server cannot offer a lease to a DHCPv4 client that is more than the MCLT beyond the current lease time for this client that is known by the failover partner. From this standpoint, it would be good for the MCLT to be as long as possible. However, in a failure situation, waiting the MCLT beyond the current lease time in order to reuse a leased lease would suggest that the MCLT should be as short as possible.

This tension is resolved by anticipating the need to extend lease times when communicating with the failover partner. The first lease offered to a DHCPv4 client can be only as long as the MCLT. However, when the failover server updates its partner, it updates the partner with the desired lease time plus the MCLT. Thus, when the client returns with a renewal request at halfway through the MCLT, the failover server can extend its lease for only the lease time known by the partner plus the MCLT. But the partner now knows the desired lease time, so that the server can extend the lease for as long as it was configured since it had pre-updated the failover partner with this time.

Using this approach, one can keep the MCLT relatively short, say 1 hour, and still offer leases of any desired extent to clients -- once the failover partner has been updated with the desired lease time.

9.1.4. Network Partition Events

It is clear that when one failover server finds that it is unable to communicate with its failover partner, it is impossible for that server to tell if its failover partner is down or if the communication path to that failover partner is broken, a situation known as "network partition" (see Section 6.1.2). The DHCPv4 failover protocol distinguishes between these different situations by having different failover states to represent "communications-interrupted" situations and a "partner-down" situations. The expectation is that (at least in some situations) it requires an operator action to distinguish between a communications-interrupted and partner-down event. In particular, the DHCPv4 failover protocol does not conflate these two situations.

Correct handling of network partition events requires that a failover server unable to communicate with its failover partner (but not yet informed that its failover partner is down), must not re-allocate an IP address from one DHCPv4 client to another. Available addresses may be allocated to any DHCPv4 client.

After a failover server has been informed that its partner is down, it can reallocate an IP address from one DHCPv4 client to another once it has waited the MCLT beyond the lease expiration of that IP address. This need to be informed by an external entity that the failover partner is down is the only impact of correctly handling network partition events. Of course, specific implementations can assume that an unreachable failover partner is down after a shorter or longer time, thus limiting the support for a network partition event.

9.1.5. Conflict Resolution

Whenever one failover server receives an update from its failover partner, it needs to decide if the update it has received is "better" than the information it has in its own database concerning the DHCPv4 client or the lease on the IPv4 address. The DHCPv4 failover protocol does not mandate the details of this decision, but this activity must be part of any DHCPv4 implementation. In most cases, comparing the times associated with the failover update with the times held in the server’s own database will allow this decision to be made.

9.1.6. Load Balancing

The DHCPv4 Load Balancing protocol [RFC3074] integrates with the DHCPv4 failover protocol by defining the way that each server decides which DHCPv4 clients to process. Use of load balancing with the

DHCPv4 failover protocol is an optional extension to the failover protocol. Both a simple active -- passive relationship without load balancing is defined as well as a more complex active -- active relationship.

9.2. DHCPv6 Redundancy Considerations

[I-D.ietf-dhc-dhcpv6-redundancy-consider] specifies an interim architecture to provide a semi-redundant DHCPv6 solution before the availability of vendor or standard based solutions. The proposed architecture may be used in wide range of networks. Two notable deployment models are discussed: service provider and enterprise network environments. The described architecture leverages only existing and implemented DHCPv6 standards.

10. DHCP Best Practices

TODO: Qin Wu provided this best practices link. Describe it briefly. http://technet.microsoft.com/en-us/library/cc780311(WS.10).aspx

11. Security Considerations

TBD...

12. IANA Considerations

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

13. Acknowledgements

This document extensively uses concepts, definitions and other parts of [dhcpv4-failover] document. Authors would like to thank Shawn Routhier, Qin Wu, Jean-Francois Tremblay, Frank Sweetser, Jiang Sheng, Yu Fu, Greg Gabil and Bernie Volz for their comments and feedback.

14. References

14.1. 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.

[RFC3074] Volz, B., Gonczi, S., Lemon, T., and R. Stevens, "DHC Load Balancing Algorithm", RFC 3074, February 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.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC4704] Volz, B., "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option", RFC 4704, October 2006.

14.2. Informative References

[I-D.ietf-dhc-dhcpv6-redundancy-consider] Brzozowski, J., Tremblay, J., Chen, J., and T. Mrugalski, "DHCPv6 Redundancy Deployment Considerations", draft-ietf-dhc-dhcpv6-redundancy-consider-00 (work in progress), April 2011.

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

[dhcpv4-failover] Droms, R., Kinnear, K., Stapp, M., Volz, B., Gonczi, S., Rabil, G., Dooley, M., and A. Kapur, "DHCP Failover Protocol", draft-ietf-dhc-failover-12 (work in progress), March 2003.

Authors' Addresses

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

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

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

Phone: +1 (978) 936-0000
Email: kkinear@cisco.com

Network Working Group
Internet-Draft
Updates: 3315 (if approved)
Intended status: Standards Track
Expires: October 19, 2011

T. Mrugalski
ISC

April 17, 2011

Requesting Suboptions in DHCPv6
draft-mrugalski-dhc-dhcpv6-suboptions-01

Abstract

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

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on October 19, 2011.

Copyright Notice

Table of Contents

1. Introduction

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

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

2. Terminology

This section defines terms used in this document.

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

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

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

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

3. Suboption Request Procedure

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

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

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

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

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

4. Justification

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

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

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

1. [I-D.ietf-dhc-pd-exclude] defines PD_EXCLUDE option that is conveyed within IAPREFIX (that in turn is conveyed in IA_PD). Currently this draft calls for requesting PD_EXCLUDE in top-level ORO.

2. [I-D.ietf-mif-dhcpv6-route-option] defines a way to convey basic information about routers and prefixes available via those routers. It defines IA_RT option that conveys NEXT_HOP option that contains RT_PREFIX options. Each of those defined options may possibly convey additional, not yet defined routing related options, e.g. MTU, flow label, QoS parameters or many others.

3. Bernie Volz mentioned on DHC mailing list that one of the vendors used vendor specific information option within existing options. That particular vendor put ORO option within their option space.

4. There is existing DHCPv6 implementation (Dibbler) that currently requests extra sub-options using top-level ORO. That is done differently than described in previous bullet. As such, interoperation is not possible at this time.

5. New draft about 4rd [I-D.despres-intarea-4rd] defines DHCPv6 option. While current definition should probably be improved, this architecture requires configuring one or more mapping rules. Each mapping rule consists of several mandatory (Domain IPv6 Prefix, Domain 4rd Prefix, Length of CE IPv6 Prefix) and one optional (Domain IPv6 suffix) parameters. As all those options are dedicated to configuration of different aspects of the same feature (4rd), there’s distinct possibility that it will be defined as several options nested within a single grouping option. Although this architecture is a new proposal, there may be new extensions proposed, similar to extensions to DS-Lite architecture. This may result in potential new options related to 4rd.

5. DHCPv6 Client Behavior

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

6. DHCPv6 Server Behavior

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

7. IANA Considerations

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

8. Security Considerations

TBD

9. References

9.1. 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.

9.2. Informative References

[I-D.despres-intarea-4rd]
Despres, R., Matsushima, S., Murakami, T., and O. Troan, "IPv4 Residual Deployment across IPv6-Service networks (4rd) ISP-NAT’s made optional", draft-despres-intarea-4rd-01 (work in progress), March 2011.

[I-D.ietf-dhc-pd-exclude]
Korhonen, J., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", draft-ietf-dhc-pd-exclude-01 (work in progress), January 2011.

[I-D.ietf-mif-dhcpv6-route-option]
Dec, W., Mrugalski, T., Sun, T., and B. Sarikaya, "DHCPv6 Route Option", draft-ietf-mif-dhcpv6-route-option-01 (work

in progress), March 2011.

Author's Address

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

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

DHC
Internet-Draft
Intended status: Standards Track
Expires: December 17, 2011

Y. Xu
S. Manning
M. Wong
Huawei Technologies
June 15, 2011

A authentication method based on certificate for DHCP
draft-xu-dhc-cadhcp-00.txt

Abstract

This document defines a technique that can provide both entity authentication and message authentication based on certificates. This protocol combines existing options, such as the delay authentication mechanism in [RFC3118] and the user authentication protocol option defined in [RFC2485]. The goal of this specification is to define methods for certificates to protect the integrity of DHCP messages and close the gaps of the existing delay authentication mechanism. In order to meet these goals, we use the asymmetrical cryptograph protection and some options about authentication that have been defined in other specifications.

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 December 11, 2011.

Copyright Notice

Table of Contents

1. Introduction

DHCP provides a framework for passing network configuration information to hosts on a TCP/IP network. Most of these parameters are IP addresses. The DHCP server can allocate addresses to clients dynamically. To ensure the security of communication between DHCP client and DHCP server, network administrators may wish to provide authentication for the DHCP clients and DHCP messages. [RFC3118] defines an authentication mechanism for DHCP, the delay authentication protocol. But it is vulnerable to denial of service attacks through flooding with DHCPDISCOVER messages, which are not authenticated by Delay authentication protocol. This attack may overwhelm the DHCP server and exhaust the addresses available for assignment by the DHCP server. Delay authentication is prone to other kinds of attacks and limitations. Further, this delay authentication is based on a pre-shared key .

This increases the overload of key distribution and management in the implementation. As defined in [RFC5280], certificates can be used in entity authentication widely. The MTU in Ethernet is usually 1500 bytes, while the certificate is usually as large as 1k or 2k bytes, and since DHCPDISCOVER messages and DHCPREQUEST messages are broadcast messages, these cannot be fragmented into several messages. Thus, directly carrying certificates in DHCP messages is impossible. This document defines a new method for Dynamic Host Configuration Protocol authentication based on certificates. The basic design philosophy is performing authentication immediately between DHCP client and DHCP server by combining some authentication options, sending URL information and the Client identity specified in [RFC2485] and [RFC2132] instead of a certificate directly, and leveraging a mechanism where the DHCP server has been authenticated by a centralized trusted server.

2. Terminology used in this document

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

3. Certificate Based Authentication

The DHCP client is configured with its certificate and the corresponding private key and the trusted server’s certificate. The DHCP server is configured with its certificate and the corresponding private key. The trusted server is configured with its certificate and the corresponding private key and certificates of DHCP clients. A DHCP client sends DHCPDISCOVER message that has been protected by its private key to DHCP server, the verification of the message is

through the DHCP server and trusted server. If successful, the DHCP server sends the DHCPOFFER message protected with the private key of the DHCP server. And the DHCP client authenticates the DHCP server by the validation of the DHCPOFFER message.

Based on the authentication option from [RFC3118], if the protocol field is 2(TBD), the message uses the certificate based authentication mechanism defined in this document. In the certificate based authentication, the client requests authentication in the DHCPDISCOVER message and the server replies with a DHCPOFFER message. Three options are included in the DHCPDISCOVER message, the Authentication Option defined in this document, the Client-identifier Option defined in [RFC2132] and the User Authentication Protocol Option defined in [RFC2485]. The DHCPDISCOVER message is signed with the private key of the DHCP client. For the Authentication Option, unlike the delayed authentication mechanism, the signature generated with the DHCP client private key is added in the Authentication Information. The Client-identifier Option (Option 61) is used to carry the DHCP client identifier. If the DHCP client is configured with a certificate, the sequence number of certificate can be used as the DHCP client identifier. The User Authentication Protocol Option is used to carry the URL of the trusted server, such as, a certificate server. The URL of the trusted server can be configured out of band.

The format of the authentication request in a DHCPDISCOVER or a DHCPINFORM message for certificate authentication is:

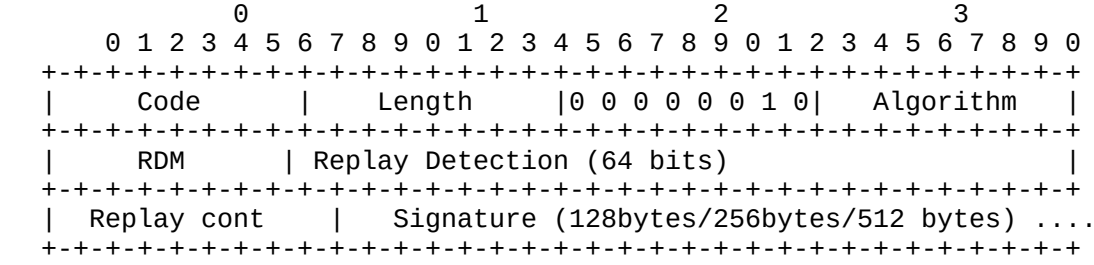


Figure 1. The format of the authentication request(DHCPDISCOVER/ DHCPINFORM)

The code for the authentication option is 90, and the length field contains the length of protocol, RDM, algorithm, Replay Detection field, and Signature. The protocol is defined to 2(00000010). The signature field is used for message validation. The other field is defined as [RFC3118] .

When the DHCP server receives the DHCPDISCOVER message, it can obtain the DHCP client's certificate by the URL and the client identity. At first, the DHCP server searches the trusted server with the URL information, and forwards the client identity information to the trusted server to obtain the DHCP client certificate. If the DHCP server is authenticated by a trusted server, the DHCP server downloads the DHCP client certificate from the trusted server. The certificate may be protected with the secure tunnel, such as, SSL/ TLS, which is established between the DHCP server and the trusted server. Through the authentication between DHCP server and the trusted server, only the legitimate DHCP server or the authenticated DHCP server can obtain the certificate of the DHCP client from the trusted server.

After the trusted server receives the client identity information, it checks the validity of the client identity. If it is legitimate, the trusted server will send the certificate to the DHCP server via the SSL/TLS tunnel. Upon receiving the DHCP client certificate, the DHCP server checks that the subject field of certificate matches with the client identity. The DHCP server validates the signature of the DHCPDISCOVER in Authentication Option. If the validation is successful, it proves that the DHCP client is in possession of the private key corresponding to the certificate. At this time, the DHCP client has been authenticated with the certificate based authentication mechanism.

4. Unicast DHCPOFFER Message

When DHCPOFFER is unicast, it can be fragmented and maybe used to carry a certificate. The DHCP server will use its private key to sign the DHCPOFFER message, which contains the configured information, such as Vendor Specific Information option, the Authentication option, and may contain other options. The certificate of the DHCP server is included in the Authentication Option. The format of the option is shown in Figure 2. If the length exceeds the MTU, it can be fragmented with several messages with same sequence number specified as in [RFC3396] . When the DHCP client receive whole the DHCPOFFER message, it obtains the DHCP server's certificate to check whether the certificate is valid, and validates the signature of the DHCPOFFER message.

The following DHCPREQUEST message and the DHCPACK message can be validated by the same authentication mechanism. The DHCP client protects the sending message with the signature generated by its private key. The DHCP server validates the signature with the public key of the DHCP client.

The format of the authentication information in a DHCPOFFER, DHCPACK message for certificate authentication is shown as follow,

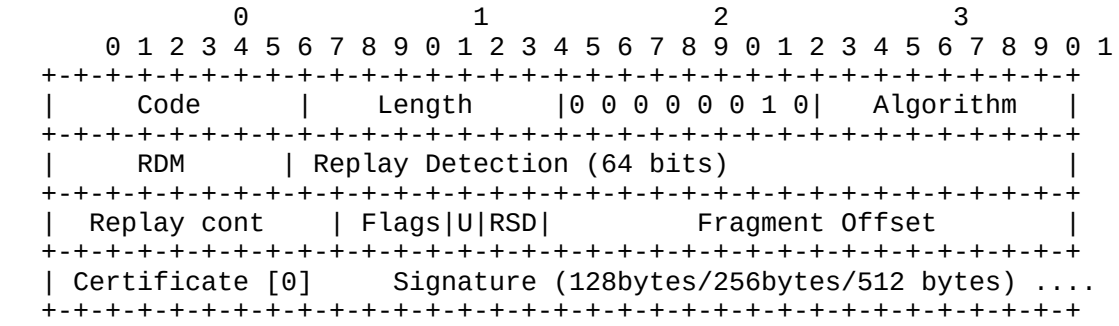


Figure 2. The format of the authentication information(DHCPOFFER/ DHCPACK)

We can use the new defined format of the option. Then we can use the following format in DHCPOFFER, DHCPACK message. And when the option exceeds 255 bytes, the method that specified in [RFC3396] will be used.

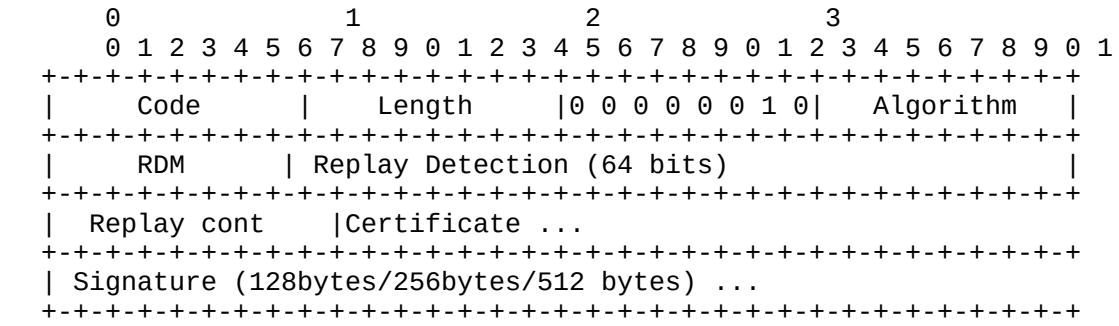


Figure 3. The format of the new authentication option

5. Authentication between DHCP server and the trusted server

The authentication mechanism between DHCP server and the trusted server may be any existing authentication method, such as, the SSL/ TLS defined in [RFC4246] and [RFC5246]. After the authentication between the trusted server and the DHCP server, the SSL/TLS tunnel is

established between DHCP server and the trusted server.

6. The Generation of signature

The signature of the message is generated through the private key and DHCP content, such as, DHCP message or some information like entity identity. For the DHCPDISCOVER and the DHCPREQUEST message, the signature is generated with the private key of the DHCP client. For the DHCPOFFER and the DHCPACK message, the signature is generated with the private key of the DHCP server and the DHCP contents.

7. Message validation

To validate the incoming DHCP messages, the receiver will check the signature with the corresponding public key. For the DHCPDISCOVER and the DHCPREQUEST message, the DHCP server first checks whether the value in replay detection field is acceptable according to the replay detection method specified by the RDM field. Next the server validates the signature with the public key in the client's certificate. For the DHCPOFFER and the DHCPACK message, the client checks the replay detection field, if it is correct, the client validates the DHCP server's certificate and checks the validity of the signature of the DHCP message to guarantee that this DHCP server has been authenticated.

8. Entity authentication

The DHCP server authenticates the DHCP client by the validation of the DHCPDISCOVER message signature. This validation is carried out by the DHCP server with the certificate of the DHCP client acquired from the trusted server. The DHCP client authenticates the DHCP server by validating the DHCP server's certificate and the signature of the DHCPOFFER message.

9. Client Considerations

This section describes the behavior of a DHCP client using the certificate based authentication.

1. The client MUST include the authentication request option using certificates where the protocol field is equal to 2 in its DHCPDISCOVER message along with the client identifier option and the User Authentication Protocol Option. The DHCPDISCOVER message MUST sign the message with the client's private key.

2. The client MUST perform the validation of the DHCP server's certificate and the signature of the DHCPOFFER message.
3. The client replies with a DHCPREQUEST message that MUST include authentication option protected by the same private key used in DHCPDISCOVER message.
4. If the client validates the DHCPOFFER it accepted, the client MUST validate the DHCPACK message from the server.

10. DHCP Server Considerations

This section describes the behavior of a DHCP server in response to client message using certificate based authentication.

1. Each server MUST be authenticated by a trusted server and can maintain the secure link with this trusted server.
2. Each server MUST validate the incoming message with the public key of the DHCP client by obtaining the certificate of the DHCP client from the trusted server.
3. Each server MUST protect the sending message by the private key of the DHCP server. If the replay detection check or the message signature validation fails, the server MUST discard the incoming message.

11. Trusted server Considerations

The trusted server is only a general name for a certificate server. Any valid authentication mechanism may be used between DHCP server and trusted server. The trusted server can regarded as high level sever that can authenticate DHCP servers.

This section describes the behavior of the trusted server using certificate based authentication.

1. The trusted server MAY authenticate the DHCP server prior to the connection between the DHCP client and DHCP server.
2. The trusted server MUST distribute the certificate of the DHCP client to a legitimate DHCP server that has been authenticated.
3. Client certificates may be cached or obtained in real time, but caching has performance gain at the expense of memory usage. As the client list grows, the DHCP server will use more memory to store the client's certificates, which will increase the overhead of certificate management. This is similar to the argument of not using PSK-based scheme.

12. Application example

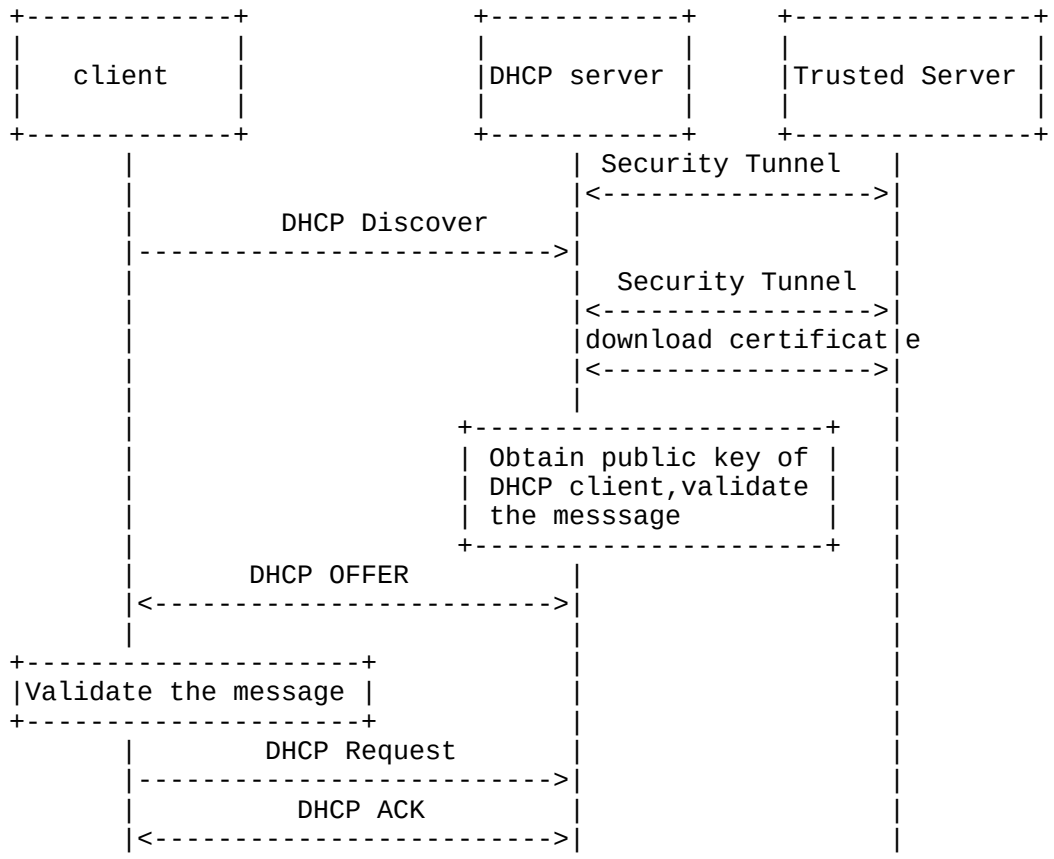


Figure 4. DHCP Example Procedure

Security tunnel will be established between DHCP server and Trust server before or after DHCP server receive DHCP DISCOVER message. With the DHCP client ID and the address information of trusted server, DHCP server obtain the corresponding public key of the DHCP client to validate the DHCP DISCOVER message. If successful, the DHCP server will send DHCPOFFER to DHCP client specified according to unicast case or broadcast case. And the client validates the DHCP OFFER corresponding to the two different cases. The authentication of following messages can be used the similar mechanism.

13. Security Considerations

1. On Signature: Signature calculation can be based on either sender's private key or receiver's public key, but with sender's private key, it has the effect of origin authentication.

2. On Authentication: The two entity authentication is considered only bi-lateral authentication and not mutual authentication. Each authentication is verified independently without both client and server contributing to the authentication. When DHCPOFFER is unicast, it can be fragmented and maybe used to carry a certificate. In this case, DHCP client may be able to receive the DHCP server's certificate. Furthermore, the DHCPOFFER may then be signed by server's private key, which also provides the benefit of origin authentication.

3. Implementations MUST support the following attribute algorithm values

a) Integrity Algorithm
i. MD5
ii. SHA1 Algorithm Type Value

RESERVED	0
HASH_MD5	1
HASH_SHA1	2
HASH_SHA256	3
HASH_SHA384	4
HASH_SHA512	5
Standards Action	6-127
Private Use	128-255
Unassigned	256-32767

b) signature algorithm
i. RSA Algorithm Type Value

RESERVED	0
RSA	1
Standards Action	2-127
Private Use	128-255
Unassigned	256-32767

14. IANA Considerations

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

15. Acknowledgments

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

16. References

16.1. 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.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

[RFC2485] Drach, S., "DHCP Option for The Open Group's User Authentication Protocol", RFC 2485, January 1999.

[RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.

[RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396, November 2002.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

16.2. Informative References

[RFC4246] Dolan, M., "International Standard Audiovisual Number (ISAN) URN Definition", RFC 4246, February 2006.

Author's Address

Yixian Xu
Huawei Technologies
Huawei Building, Xinxi Road No.3
Haidian District, Beijing 100085
P. R. China

Phone: +86-10-82836300
Email: xuyixian@huawei.com

Serge Manning
Huawei Technologies

Phone: 001-9725435324
Email: serge.manning@huawei.com

Marcus Wong
Huawei Technologies

Phone: 001-908-5413505
Email: mwong@huawei.com

DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 29, 2011

L. Yeh, Ed.
T. Tsou
Huawei Technologies
M. Boucadair
France Telecom
J. Hu
China Telecom
June 27, 2011

Prefix Pool Option for DHCPv6 Relay Agent on Provider Edge Router
draft-yeh-dhc-dhcpv6-prefix-pool-opt-04

Abstract

The Prefix Pool option provides an automatic mechanism based on the DHCPv6-PD, allowing the DHCPv6 server to notify the DHCPv6 relay agent implemented only on the Provider Edge (PE) router about the information of prefix pools. The information of prefix pools can be used to enforce IPv6 route aggregation on the PE router by adding or removing the aggregated routes per the status of the prefix pools. The advertising of the aggregated routes in the routing protocol enabled on the network-facing interface of PE router will dramatically decreases the number of the routes 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 December 29, 2011.

Copyright Notice

Table of Contents

1. Introduction

DHCPv6 [RFC3315] protocol specifies a mechanism for the assignment of IPv6 address and configuration information to IPv6 nodes. 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 Router (RR) acting as the DHCPv6 Client. DHCPv6 servers always maintain authoritative information related to their operations including, but not limited to, binding data of the delegated IPv6 prefixes, lease data of the delegated IPv6 prefixes, and binding data of their prefix pools, etc.

In the scenario of the centralized DHCP server, the Provider Edge (PE) routers act as a DHCPv6 relay agents when the DHCPv6 Server acting as DR and the DHCPv6 clients acting as RRs are not on the same link. For the reachability purpose, 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 customer routers (a.k.a. Routed-RG or Routed-CPE), which acts as RRs. (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 to be unacceptable huge, so that it is necessary to aggregate the routes directing to the customer networks on the PE router.

Because the prefixes of the customer networks can not guarantee always to be valid and continuous, the routing protocol enabled on the PE router can not make one aggregated route automatically to cover all the prefixes delegated within the prefix pool. The PE router needs other ways to make the aggregated routes. One way to make the aggregated routes is to configure them manually and permanently per the provision of the prefix pools, but PE router doesn’t know the information about the prefix pools when it acts as the relay agent.

This document proposes a new Prefix Pool option for the DHCPv6 reply agent implemented only on the PE routers, allowing the server to notify the relay agent about the information of prefix pools. After the PE router got the information of the prefix pools, the aggregated route entries (e.g., black-hole routes) pointing to each of the prefix pools can be added or withdrawn in the routing table of the PE router. The aggregated routes will 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. DHCPv6 relay agent), to support the replacement or reboot event of relay agent. In this document, the capability of DHCPv6 Bulk Leasequery will be extended to support the bulk transfer of the binding data of prefix pool for the route aggregation.

2. Terminology and Language

This document describes new DHCPv6 options of prefix pool and the associated mechanism on the relay agent and server. This document should be read in conjunction with the DHCPv6 specification, RFC3315, RFC3633, RFC5007 and RFC5460 for a complete mechanism. Definitions for terms and acronyms not specifically defined in this document are defined in RFC3315, RFC3633, [RFC3769], [RFC5007] and RFC5460.

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.

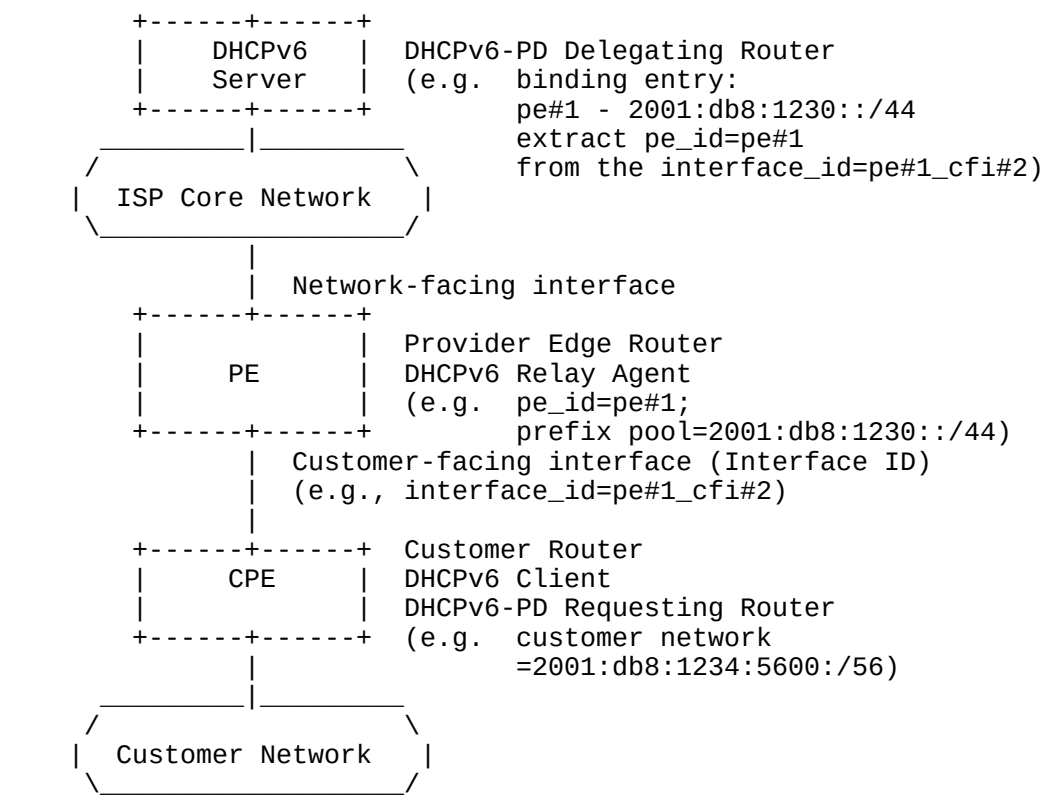


Figure 1: Use case of ISP-Customer network where CPE is directly connected to PE

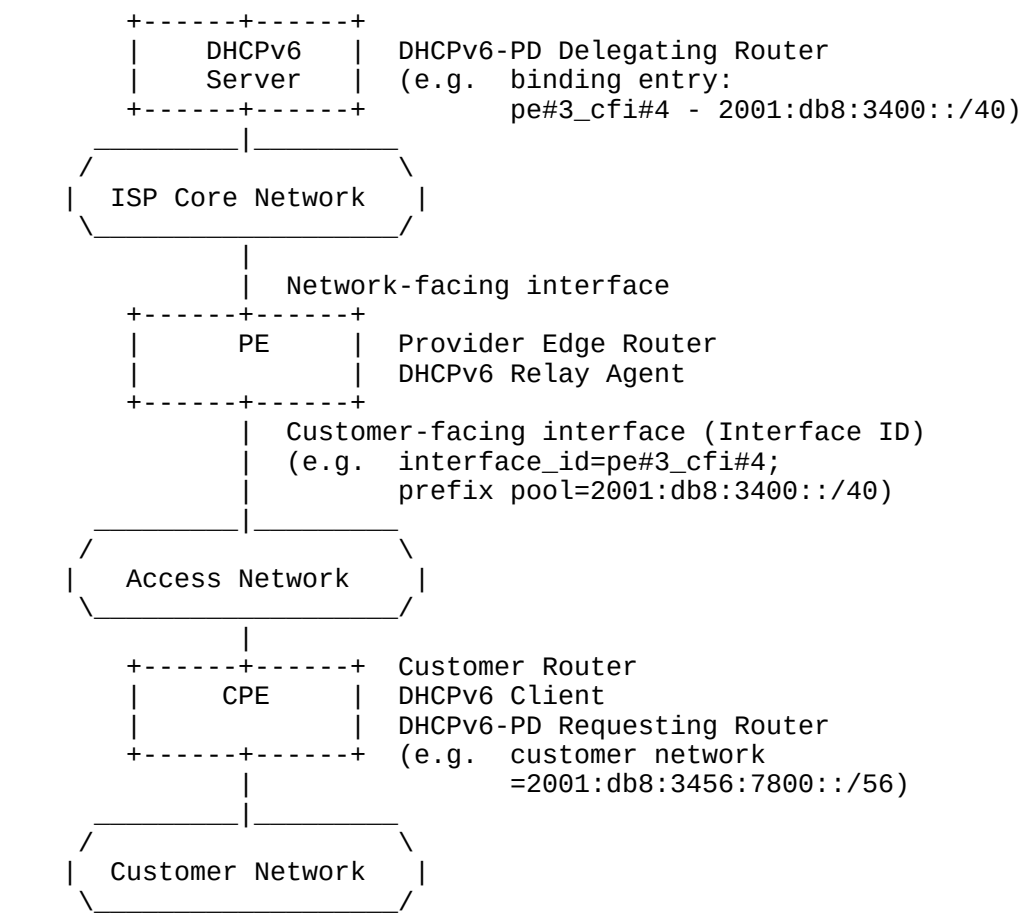
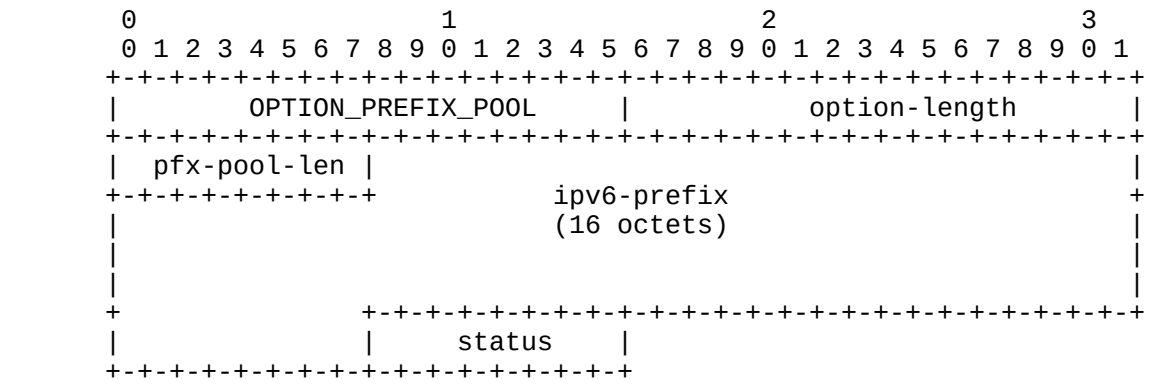


Figure 2: Use case of ISP-Customer network where CPE is connected to PE through access network

4. Prefix Pool Option

The format of the Prefix Pool option is shown is 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
Valid	0
Released	1
Reserved	2˜255

The status of 'Valid' in the Prefix Pool option can be used to add the prefix pool and the associated aggregated route on the relay agent; while the status of 'Released' in the Prefix Pool option can be used to withdraw the prefix pool and the associated aggregated route on the relay agent.

If the administrative policy on the server permit and support the 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 valid lease, the status of prefix pool will be 'Valid'. Otherwise, the status of prefix pool is 'Released'. If the administrative policy on the server don't permit or support the route aggregation on the relay agent, the status of prefix pool will always be 'Released'.

5. Relay Agent Behavior

The relay agent who needs the information of prefix pools, must

includes Option Request option (OPTION_ORO, 6) to request Prefix Pool option from the 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) where receiving the DHCPv6-PD message from clients. The relay agent can include this Option Request option for 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 Prefix Pool option with the field values of pfx-pool-len and IPv6-prefix as its preference which the relay agent would like the server to return.

The relay agent should include Interface ID option (OPTION_INTERFACE_ID, 18) for the server to identify the relay agent itself or its particular customer-facing interface where the prefix pool is associated, if the server would not like to use link-address specified in the DHCPv6 message encapsulation of relay-forward message to identify the interface of the link on which the clients are located.

After received 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 server, the relay agent shall add or withdraw the aggregated route entry per the status of the prefix pool. If the status of the prefix pool got from the server is 'Valid', the relay agent shall add an aggregated route entry in its routing table, if the same entry has not been added in. If the status of the prefix pool got from the server is 'Released', the relay agent shall withdraw the associated aggregated route entry in its routing table, if the same entry has not been withdrawn. If there is no route entry directing to the customer network within the associated aggregated route, the relay agent shall automatically withdraw the aggregated route.

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 DHCPv6 Bulk Leasequery [RFC5460] to query the binding data of prefix pools in the 'Valid' 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 Prefix Pool option from the server.

6. Server Behavior

Per DHCPv6-PD [RFC3633], if the prefix of the customer network requested in relay-forward message of SOLICIT , REQUEST, RENEW, REBIND by the Client (i.e. RR) has valid lease, the Server (i.e. DR) will delegate the prefix with the relevant parameters in the relay-reply message of REPLY. In order to give a meaningful reply, the server has to be able to maintains the binding data of the delegated IPv6 prefixes with the identification of the client. 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) to request Prefix Pool option in the relay-forward message of PD, the server must include 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 message of PD if the relay-forward message of PD received is 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 got from the relay-forward message into the relay-reply message.

If the server is set to support the route aggregation on the relay agent for the particular prefix pool, the status of this prefix pool can be determined by the delegated prefixes within the associated prefix pool. If at least one of delegated prefix in the associated prefix pool has valid lease, the server shall set the status of the prefix pool to be 'Valid'. If the lease of each delegated prefix within the associated prefix pool got expired, or if the delegated prefix in the relay-forward message of RELEASE is 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 server is set to not support the route aggregation on the relay agent for

the particular prefix pool, the status of prefix pool will always be 'Released'.

When the administrator of the server changes the setting to support the route aggregation on the relay agent for the particular prefix pool or not, the server may initiates the relay-reply message of RECONFIGURE (10) including Prefix Pool option to add or withdraw the prefix pool and the associated aggregated route on the relay agent if at least one delegated prefix within the prefix pool still has the valid lease.

Note that multiple prefix pools may associate with the same PE router implementing relay agent (Figure 1) or its customer-facing interface (Figure 2) in the binding table on the server, and the delegated prefix is only from one prefix pool.

After received the LEASEQUERY (14) message from the relay agent with the Query option (OPTION_LQ_QUERY, 44) including Option Request option (OPTION_ORO, 6) to request Prefix Pool option, 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 'Valid' status through the established TCP connection per RFC5460. Each Client Data option shall contain Prefix Pool option, and might contain the Interface ID option. In order to be able to give the meaningful replies to different type of query, 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 RFC 3315.

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. Acknowledgements

Thanks to the DHC working group members, Bernie Volz, Eliot Lear, Ole

Troan, Roberta Maglione, Ted Lemon, for the discussion in the mailing list. Thanks 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. Thanks to Sven Ooghe for some revison after the email review. Thanks to Shrinivas Ashok Joshi for pointing out the draft should cover the robust mechanism against the case of reboot.

10. Changes Log

If this document is accepted for publication as an RFC, this change log is to be removed before publication.

Rev. -04

a. Re-titled the draft to emphasize that the new mechanism with DHCPv6-PD is only designed for the PE router.

b. Re-write the abstract and some words in the introduction.

Rev. -03

a. Revisions on the behavior of Relay Agent about the automatic withdrawal of the aggregated route.

b. Re-correct the behavior of Server about the Interface ID option.

Rev. -02

a. Add one more use case of ISP network architecture where CPE is directly connected to PE.

b. Revisions on the usage of the ’status’ field in Prefix Pool option.

c. Extend DHCPv6 Bulk Leasequery (RFC5460) for the new usage.

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.

[RFC3769] Miyakawa, S. and R. Droms, "Requirements for IPv6 Prefix Delegation", RFC 3769, June 2004.

[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.

Authors' Addresses

Leaf Y. Yeh (editor)
Huawei Technologies
Area F, Huawei Park, Bantian,
Longgang District, Shenzhen 518129
P.R.China

Phone: +86-755-28971871
Email: leaf.y.yeh@huawei.com

Tina Tsou
Huawei Technologies
USA

Email: tena@huawei.com

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

Jie Hu
China Telecom
No.118, Xi Zhi Men-Nei Da Jie,
Xicheng District, Beijing 100035
P.R.China

Phone: +86-10-58552808
Email: huj@ctbri.com.cn