

Dynamic Host Configuration (DHC)  
Internet-Draft  
Obsoletes: [3315](#), 3633, 3736, 7083, 7550 (if  
approved)  
Intended status: Standards Track  
Expires: April 21, 2016

T. Mrugalski, Ed.  
M. Siodelski  
ISC  
B. Volz  
A. Yourtchenko  
Cisco  
M. Richardson  
SSW  
S. Jiang  
Huawei  
T. Lemon  
Nominum  
October 19, 2015

**Dynamic Host Configuration Protocol for IPv6 (DHCPv6) bis  
draft-ietf-dhc-rfc3315bis-02**

Abstract

The Dynamic Host Configuration Protocol for IPv6 (DHCP) enables DHCP servers to pass configuration parameters such as IPv6 network addresses to IPv6 nodes. It offers the capability of automatic allocation of reusable network addresses and additional configuration flexibility. This protocol is a stateful counterpart to "IPv6 Stateless Address Autoconfiguration" ([RFC 4862](#)), and can be used separately or concurrently with the latter to obtain configuration parameters.

This document updates the original DHCPv6 specification ([RFC 3315](#)) and incorporates [RFC 3633](#), [RFC 3736](#), [RFC 7083](#), and [RFC 7550](#).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on April 21, 2016.

## Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction and Overview</a>	<a href="#">6</a>
<a href="#">1.1.</a>	<a href="#">Relation to previous DHCPv6 standards</a>	<a href="#">6</a>
<a href="#">1.2.</a>	<a href="#">Relation to DHCP in IPv4</a>	<a href="#">7</a>
<a href="#">1.3.</a>	<a href="#">Protocols and Addressing</a>	<a href="#">7</a>
<a href="#">1.4.</a>	<a href="#">Client-server Exchanges Involving Two Messages</a>	<a href="#">8</a>
<a href="#">1.5.</a>	<a href="#">Client-server Exchanges Involving Four Messages</a>	<a href="#">8</a>
<a href="#">2.</a>	<a href="#">Requirements</a>	<a href="#">9</a>
<a href="#">3.</a>	<a href="#">Background</a>	<a href="#">9</a>
<a href="#">4.</a>	<a href="#">Terminology</a>	<a href="#">10</a>
<a href="#">4.1.</a>	<a href="#">IPv6 Terminology</a>	<a href="#">10</a>
<a href="#">4.2.</a>	<a href="#">DHCP Terminology</a>	<a href="#">11</a>
<a href="#">5.</a>	<a href="#">Operational Models</a>	<a href="#">15</a>
<a href="#">5.1.</a>	<a href="#">Stateless DHCP</a>	<a href="#">15</a>
<a href="#">5.2.</a>	<a href="#">DHCP for Non-Temporary Address Assignment</a>	<a href="#">15</a>
<a href="#">5.3.</a>	<a href="#">DHCP for Prefix Delegation</a>	<a href="#">16</a>
<a href="#">5.4.</a>	<a href="#">DHCP for Customer Edge Routers</a>	<a href="#">18</a>
<a href="#">5.5.</a>	<a href="#">DHCP for Temporary Addresses</a>	<a href="#">19</a>
<a href="#">6.</a>	<a href="#">DHCP Constants</a>	<a href="#">19</a>
<a href="#">6.1.</a>	<a href="#">Multicast Addresses</a>	<a href="#">19</a>



6.2.	UDP Ports . . . . .	19
6.3.	DHCP Message Types . . . . .	19
6.4.	Status Codes . . . . .	21
6.5.	Transmission and Retransmission Parameters . . . . .	22
6.6.	Representation of time values and "Infinity" as a time value . . . . .	22
7.	Client/Server Message Formats . . . . .	22
8.	Relay Agent/Server Message Formats . . . . .	23
8.1.	Relay-forward Message . . . . .	24
8.2.	Relay-reply Message . . . . .	25
9.	Representation and Use of Domain Names . . . . .	25
10.	DHCP Unique Identifier (DUID) . . . . .	25
10.1.	DUID Contents . . . . .	26
10.2.	DUID Based on Link-layer Address Plus Time, DUID-LLT . .	26
10.3.	DUID Assigned by Vendor Based on Enterprise Number, DUID-EN . . . . .	28
10.4.	DUID Based on Link-layer Address, DUID-LL . . . . .	29
11.	Identity Association . . . . .	30
11.1.	Identity Associations for Address Assignment . . . . .	30
11.2.	Identity Associations for Prefix Delegation . . . . .	30
12.	Selecting Addresses for Assignment to an IA . . . . .	31
13.	Management of Temporary Addresses . . . . .	32
14.	Transmission of Messages by a Client . . . . .	33
14.1.	Rate Limiting . . . . .	33
14.2.	Choosing transmission time when T1,T2 is 0 . . . . .	34
15.	Reliability of Client Initiated Message Exchanges . . . . .	34
16.	Message Validation . . . . .	36
16.1.	Use of Transaction IDs . . . . .	36
16.2.	Solicit Message . . . . .	37
16.3.	Advertise Message . . . . .	37
16.4.	Request Message . . . . .	37
16.5.	Confirm Message . . . . .	37
16.6.	Renew Message . . . . .	38
16.7.	Rebind Message . . . . .	38
16.8.	Decline Messages . . . . .	38
16.9.	Release Message . . . . .	38
16.10.	Reply Message . . . . .	39
16.11.	Reconfigure Message . . . . .	39
16.12.	Information-request Message . . . . .	40
16.13.	Relay-forward Message . . . . .	40
16.14.	Relay-reply Message . . . . .	40
17.	Client Source Address and Interface Selection . . . . .	40
17.1.	Address Assignment . . . . .	40
17.2.	Prefix Delegation . . . . .	41
18.	DHCP Server Solicitation . . . . .	41
18.1.	Client Behavior . . . . .	42
18.1.1.	Creation of Solicit Messages . . . . .	42
18.1.2.	Transmission of Solicit Messages . . . . .	43



18.1.3.	Receipt of Advertise Messages . . . . .	44
18.1.4.	Receipt of Reply Message . . . . .	45
18.2.	Server Behavior . . . . .	46
18.2.1.	Receipt of Solicit Messages . . . . .	46
18.2.2.	Creation and Transmission of Advertise Messages . . . . .	46
18.2.3.	Creation and Transmission of Reply Messages . . . . .	48
19.	DHCP Client-Initiated Configuration Exchange . . . . .	48
19.1.	Client Behavior . . . . .	49
19.1.1.	Creation and Transmission of Request Messages . . . . .	49
19.1.2.	Creation and Transmission of Confirm Messages . . . . .	50
19.1.3.	Creation and Transmission of Renew Messages . . . . .	52
19.1.4.	Creation and Transmission of Rebind Messages . . . . .	53
19.1.5.	Creation and Transmission of Information-request Messages . . . . .	54
19.1.6.	Creation and Transmission of Release Messages . . . . .	55
19.1.7.	Creation and Transmission of Decline Messages . . . . .	56
19.1.8.	Receipt of Reply Messages . . . . .	57
19.1.9.	Bindings Verification by Requesting Router . . . . .	60
19.2.	Server Behavior . . . . .	60
19.2.1.	Receipt of Request Messages . . . . .	61
19.2.2.	Receipt of Confirm Messages . . . . .	62
19.2.3.	Receipt of Renew Messages . . . . .	62
19.2.4.	Receipt of Rebind Messages . . . . .	64
19.2.5.	Receipt of Information-request Messages . . . . .	66
19.2.6.	Receipt of Release Messages . . . . .	67
19.2.7.	Receipt of Decline Messages . . . . .	67
19.2.8.	Transmission of Reply Messages . . . . .	68
20.	DHCP Server-Initiated Configuration Exchange . . . . .	68
20.1.	Server Behavior . . . . .	69
20.1.1.	Creation and Transmission of Reconfigure Messages . . . . .	69
20.1.2.	Time Out and Retransmission of Reconfigure Messages . . . . .	70
20.2.	Receipt of Renew or Rebind Messages . . . . .	70
20.3.	Receipt of Information-request Messages . . . . .	70
20.4.	Client Behavior . . . . .	71
20.4.1.	Receipt of Reconfigure Messages . . . . .	71
20.4.2.	Creation and Transmission of Renew or Rebind Messages . . . . .	72
20.4.3.	Creation and Transmission of Information-request Messages . . . . .	72
20.4.4.	Time Out and Retransmission of Renew, Rebind or Information-request Messages . . . . .	72
20.4.5.	Receipt of Reply Messages . . . . .	72
20.5.	Prefix Delegation Reconfiguration . . . . .	73
20.5.1.	Delegating Router Behavior . . . . .	73
20.5.2.	Requesting Router Behavior . . . . .	73
21.	Relay Agent Behavior . . . . .	73
21.1.	Relaying a Client Message or a Relay-forward Message . . . . .	73
21.1.1.	Relaying a Message from a Client . . . . .	74



<a href="#">21.1.2.</a>	Relaying a Message from a Relay Agent . . . . .	<a href="#">74</a>
<a href="#">21.1.3.</a>	Relay Agent Behavior with Prefix Delegation . . . . .	<a href="#">75</a>
<a href="#">21.2.</a>	Relaying a Relay-reply Message . . . . .	<a href="#">75</a>
<a href="#">21.3.</a>	Construction of Relay-reply Messages . . . . .	<a href="#">75</a>
<a href="#">22.</a>	Authentication of DHCP Messages . . . . .	<a href="#">76</a>
22.1.	Security of Messages Sent Between Servers and Relay Agents . . . . .	<a href="#">77</a>
<a href="#">22.2.</a>	Summary of DHCP Authentication . . . . .	<a href="#">78</a>
<a href="#">22.3.</a>	Replay Detection . . . . .	<a href="#">79</a>
<a href="#">22.4.</a>	Reconfigure Key Authentication Protocol . . . . .	<a href="#">79</a>
22.4.1.	Use of the Authentication Option in the Reconfigure Key Authentication Protocol . . . . .	<a href="#">79</a>
22.4.2.	Server considerations for Reconfigure Key protocol .	80
22.4.3.	Client considerations for Reconfigure Key protocol .	81
<a href="#">23.</a>	DHCP Options . . . . .	<a href="#">81</a>
<a href="#">23.1.</a>	Format of DHCP Options . . . . .	<a href="#">81</a>
<a href="#">23.2.</a>	Client Identifier Option . . . . .	<a href="#">82</a>
<a href="#">23.3.</a>	Server Identifier Option . . . . .	<a href="#">83</a>
23.4.	Identity Association for Non-temporary Addresses Option	83
<a href="#">23.5.</a>	Identity Association for Temporary Addresses Option . .	<a href="#">86</a>
<a href="#">23.6.</a>	IA Address Option . . . . .	<a href="#">87</a>
<a href="#">23.7.</a>	Option Request Option . . . . .	<a href="#">89</a>
<a href="#">23.8.</a>	Preference Option . . . . .	<a href="#">90</a>
<a href="#">23.9.</a>	Elapsed Time Option . . . . .	<a href="#">90</a>
<a href="#">23.10.</a>	Relay Message Option . . . . .	<a href="#">91</a>
<a href="#">23.11.</a>	Authentication Option . . . . .	<a href="#">92</a>
<a href="#">23.12.</a>	Server Unicast Option . . . . .	<a href="#">93</a>
<a href="#">23.13.</a>	Status Code Option . . . . .	<a href="#">94</a>
<a href="#">23.14.</a>	Rapid Commit Option . . . . .	<a href="#">96</a>
<a href="#">23.15.</a>	User Class Option . . . . .	<a href="#">97</a>
<a href="#">23.16.</a>	Vendor Class Option . . . . .	<a href="#">98</a>
<a href="#">23.17.</a>	Vendor-specific Information Option . . . . .	<a href="#">100</a>
<a href="#">23.18.</a>	Interface-Id Option . . . . .	<a href="#">102</a>
<a href="#">23.19.</a>	Reconfigure Message Option . . . . .	<a href="#">103</a>
<a href="#">23.20.</a>	Reconfigure Accept Option . . . . .	<a href="#">103</a>
<a href="#">23.21.</a>	Identity Association for Prefix Delegation Option . . .	<a href="#">104</a>
<a href="#">23.22.</a>	IA Prefix Option . . . . .	<a href="#">106</a>
<a href="#">23.23.</a>	SOL_MAX_RT Option . . . . .	<a href="#">107</a>
<a href="#">23.24.</a>	INF_MAX_RT Option . . . . .	<a href="#">108</a>
<a href="#">24.</a>	Security Considerations . . . . .	<a href="#">110</a>
<a href="#">25.</a>	IANA Considerations . . . . .	<a href="#">112</a>
<a href="#">26.</a>	Obsoleted mechanisms . . . . .	<a href="#">112</a>
<a href="#">27.</a>	Acknowledgments . . . . .	<a href="#">113</a>
<a href="#">28.</a>	References . . . . .	<a href="#">114</a>
<a href="#">28.1.</a>	Normative References . . . . .	<a href="#">114</a>
<a href="#">28.2.</a>	Informative References . . . . .	<a href="#">116</a>
<a href="#">Appendix A.</a>	Changes since <a href="#">RFC3315</a> . . . . .	<a href="#">118</a>
<a href="#">Appendix B.</a>	Changes since <a href="#">RFC3633</a> . . . . .	<a href="#">122</a>





<a href="#">Appendix C</a> . Appearance of Options in Message Types . . . . .	<a href="#">122</a>
<a href="#">Appendix D</a> . Appearance of Options in the Options Field of DHCP Options . . . . .	<a href="#">123</a>
Authors' Addresses . . . . .	<a href="#">124</a>

## **1. Introduction and Overview**

This document describes DHCP for IPv6 (DHCPv6), a client/server protocol that provides managed configuration of devices. Relay agent functionality is also defined for enabling communication between clients and servers that are not on the same link.

DHCPv6 can provide a device with addresses assigned by a DHCPv6 server and other configuration information, which are carried in options. DHCPv6 can be extended through the definition of new options to carry configuration information not specified in this document.

DHCPv6 is the "stateful address autoconfiguration protocol" and the "stateful autoconfiguration protocol" referred to in "IPv6 Stateless Address Autoconfiguration" [[RFC4862](#)].

This document also provides a mechanism for automated delegation of IPv6 prefixes using DHCPv6. Through this mechanism, a delegating router can delegate prefixes to requesting routers.

DHCPv6 can also operate in mode, where only configuration options (but no addresses or prefixes) are provided. That implies that the server doesn't have to track any state, and thus the mode is called stateless DHCPv6. Mechanisms necessary to support stateless DHCPv6 are much smaller than to support stateful DHCPv6.

The remainder of this introduction summarizes relation to the previous DHCPv6 standards [Section 1.1](#), clarifies the stance with regards to DHCPv4 [Section 1.2](#). More detailed description of the message exchange mechanisms and example message flows in [Section 1.4](#) and [Section 1.5](#) are intended as illustrations of DHCP operation rather than an exhaustive list of all possible client-server interactions. [Section 5](#) provides an overview of common operational models. [Section 18](#), [Section 19](#), and [Section 20](#) explain client and server operation in detail.

### **[1.1](#). Relation to previous DHCPv6 standards**

The initial specification of DHCPv6 was defined in [[RFC3315](#)] and a number of follow up extensions published over the years. Several notable extensions were published: prefix delegation [[RFC3633](#)], stateless [[RFC3736](#)], update to SOL\_MAX\_RT and INF\_MAX\_RT option



values [[RFC7083](#)] and harmonization between addresses and prefixes support [[RFC7550](#)]. Understanding a protocol which definition is spread between large number of documents may be cumbersome. Furthermore, a significant operational experience has been gained over the years and certain small elements of the protocol have been reworked. This document provides a unified, corrected and cleaned up definition of the DHCPv6 that also covers all erratas filled against older RFCs. As such, it obsoletes a number of aforementioned RFCs. There is a small number of mechanisms that were obsoleted. They are listed in [Section 26](#).

## **[1.2.](#) Relation to DHCP in IPv4**

The operational models and relevant configuration information for DHCPv4 [[RFC2132](#)][RFC2131] and DHCPv6 are sufficiently different that integration between the two services is not included in this document. [[RFC3315](#)] suggested that future work might be to extend DHCPv6 to carry IPv4 address and configuration information. However, the current consensus of the IETF is that DHCPv4 should be used rather than DHCPv6 when conveying IPv4 configuration information to nodes. [[RFC7341](#)] describes a transport mechanism to carry DHCPv4 messages using the DHCPv6 protocol for the dynamic provisioning of IPv4 address and configuration information across IPv6-only networks.

## **[1.3.](#) Protocols and Addressing**

Clients and servers exchange DHCP messages using UDP [[RFC0768](#)]. The client uses a link-local address or addresses determined through other mechanisms for transmitting and receiving DHCP messages.

A DHCP client sends most messages using a reserved, link-scoped multicast destination address so that the client need not be configured with the address or addresses of DHCP servers.

To allow a DHCP client to send a message to a DHCP server that is not attached to the same link, a DHCP relay agent on the client's link will relay messages between the client and server. The operation of the relay agent is transparent to the client and the discussion of message exchanges in the remainder of this section will omit the description of message relaying by relay agents.

Once the client has determined the address of a server, it may under some circumstances send messages directly to the server using unicast.



#### **1.4. Client-server Exchanges Involving Two Messages**

When a DHCP client does not need to have a DHCP server assign it IP addresses, the client can obtain configuration information such as a list of available DNS servers [[RFC3646](#)] or NTP servers [[RFC4075](#)] through a single message and reply exchanged with a DHCP server. To obtain configuration information the client first sends an Information-request message to the All\_DHCP\_Relay\_Agents\_and\_Servers multicast address. Servers respond with a Reply message containing the configuration information for the client.

This message exchange assumes that the client requires only configuration information and does not require the assignment of any IPv6 addresses.

When a server has IPv6 addresses and other configuration information committed to a client, the client and server may be able to complete the exchange using only two messages, instead of four messages as described in the next section. In this case, the client sends a Solicit message to the All\_DHCP\_Relay\_Agents\_and\_Servers requesting the assignment of addresses and other configuration information. This message includes an indication that the client is willing to accept an immediate Reply message from the server. The server that is willing to commit the assignment of addresses to the client immediately responds with a Reply message. The configuration information and the addresses in the Reply message are then immediately available for use by the client.

Each address assigned to the client has associated preferred and valid lifetimes specified by the server. To request an extension of the lifetimes assigned to an address, the client sends a Renew message to the server. The server sends a Reply message to the client with the new lifetimes, allowing the client to continue to use the address without interruption.

#### **1.5. Client-server Exchanges Involving Four Messages**

To request the assignment of one or more IPv6 addresses, a client first locates a DHCP server and then requests the assignment of addresses and other configuration information from the server. The client sends a Solicit message to the All\_DHCP\_Relay\_Agents\_and\_Servers address to find available DHCP servers. Any server that can meet the client's requirements responds with an Advertise message. The client then chooses one of the servers and sends a Request message to the server asking for confirmed assignment of addresses and other configuration information. The server responds with a Reply message that contains the confirmed addresses and configuration.



As described in the previous section, the client sends a Renew message to the server to extend the lifetimes associated with its addresses, allowing the client to continue to use those addresses without interruption.

## **2. Requirements**

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

This document also makes use of internal conceptual variables to describe protocol behavior and external variables that an implementation must allow system administrators to change. The specific variable names, how their values change, and how their settings influence protocol behavior are provided to demonstrate protocol behavior. An implementation is not required to have them in the exact form described here, so long as its external behavior is consistent with that described in this document.

## **3. Background**

The IPv6 Specification provides the base architecture and design of IPv6. Related work in IPv6 that would best serve an implementor to study includes the IPv6 Specification [[RFC2460](#)], the IPv6 Addressing Architecture [[RFC4291](#)], IPv6 Stateless Address Autoconfiguration [[RFC4862](#)], IPv6 Neighbor Discovery Processing [[RFC4861](#)], and Dynamic Updates to DNS [[RFC2136](#)]. These specifications enable DHCP to build upon the IPv6 work to provide both robust stateful autoconfiguration and autoregistration of DNS Host Names.

The IPv6 Addressing Architecture specification [[RFC4291](#)] defines the address scope that can be used in an IPv6 implementation, and the various configuration architecture guidelines for network designers of the IPv6 address space. Two advantages of IPv6 are that support for multicast is required and nodes can create link-local addresses during initialization. The availability of these features means that a client can use its link-local address and a well-known multicast address to discover and communicate with DHCP servers or relay agents on its link.

IPv6 Stateless Address Autoconfiguration [[RFC4862](#)] specifies procedures by which a node may autoconfigure addresses based on router advertisements [[RFC4861](#)], and the use of a valid lifetime to support renumbering of addresses on the Internet. In addition, the protocol interaction by which a node begins stateless or stateful autoconfiguration is specified. DHCP is one vehicle to perform





stateful autoconfiguration. Compatibility with stateless address autoconfiguration is a design requirement of DHCP.

IPv6 Neighbor Discovery [[RFC4861](#)] is the node discovery protocol in IPv6 which replaces and enhances functions of ARP [[RFC0826](#)]. To understand IPv6 and stateless address autoconfiguration, it is strongly recommended that implementors understand IPv6 Neighbor Discovery.

Dynamic Updates to DNS [[RFC2136](#)] is a specification that supports the dynamic update of DNS records for both IPv4 and IPv6. DHCP can use the dynamic updates to DNS to integrate addresses and name space to not only support autoconfiguration, but also autoregistration in IPv6.

## **[4.](#) Terminology**

This section defines terminology specific to IPv6 and DHCP used in this document.

### **[4.1.](#) IPv6 Terminology**

IPv6 terminology relevant to this specification from the IPv6 Protocol [[RFC2460](#)], IPv6 Addressing Architecture [[RFC4291](#)], and IPv6 Stateless Address Autoconfiguration [[RFC4862](#)] is included below.

address	An IP layer identifier for an interface or a set of interfaces.
host	Any node that is not a router.
IP	Internet Protocol Version 6 (IPv6). The terms IPv4 and IPv6 are used only in contexts where it is necessary to avoid ambiguity.
interface	A node's attachment to a link.
link	A communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IP. Examples are Ethernet (simple or bridged); Token Ring; PPP links, X.25, Frame Relay, or ATM networks; and Internet (or higher) layer "tunnels", such as tunnels over IPv4 or IPv6 itself.



link-layer identifier	A link-layer identifier for an interface. Examples include IEEE 802 addresses for Ethernet or Token Ring network interfaces, and E.164 addresses for ISDN links.
link-local address	An IPv6 address having a link-only scope, indicated by having the prefix (FE80::/10), that can be used to reach neighboring nodes attached to the same link. Every interface has a link-local address.
multicast address	An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.
neighbor	A node attached to the same link.
node	A device that implements IP.
packet	An IP header plus payload.
prefix	The initial bits of an address, or a set of IP addresses that share the same initial bits.
prefix length	The number of bits in a prefix.
router	A node that forwards IP packets not explicitly addressed to itself.
unicast address	An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.

#### [4.2.](#) DHCP Terminology

Terminology specific to DHCP can be found below.

appropriate to the link	An address is "appropriate to the link" when the address is consistent with the DHCP server's knowledge of the network topology, prefix assignment and address assignment policies.
-------------------------	---



binding	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. A binding containing information about an IA is indexed by the tuple <DUID, IA-type, IAID> (where IA-type is the type of address in the IA; for example, temporary). A binding containing configuration information for a client is indexed by <DUID>.
configuration parameter	An element of the configuration information set on the server and delivered to the client using DHCP. Such parameters may be used to carry information to be used by a node to configure its network subsystem and enable communication on a link or internetwork, for example.
delegating router	The router that acts as a DHCP server, and is responding to the prefix request.
DHCP	Dynamic Host Configuration Protocol for IPv6. The terms DHCPv4 and DHCPv6 are used only in contexts where it is necessary to avoid ambiguity.
DHCP client (or client)	A node that initiates requests on a link to obtain configuration parameters from one or more DHCP servers. Depending on the purpose of the client, it may feature the requesting router functionality, if it supports prefix delegation.
DHCP domain	A set of links managed by DHCP and operated by a single administrative entity.
DHCP realm	A name used to identify the DHCP administrative domain from which a DHCP authentication key was selected.



DHCP relay agent (or relay agent)	A node that acts as an intermediary to deliver DHCP messages between clients and servers. In certain configurations there may be more than one relay agent between clients and servers, so a relay agent may send DHCP messages to another relay agent.
DHCP server (or server)	A node that responds to requests from clients, and may or may not be on the same link as the client(s). Depending on its capabilities, it may also feature the functionality of delegating router, if it supports prefix delegation.
DUID	A DHCP Unique IDentifier for a DHCP participant; each DHCP client and server has exactly one DUID. See <a href="#">Section 10</a> for details of the ways in which a DUID may be constructed.
IA	Identity Association: A collection of leases assigned to a client. Each IA has an associated IAID. A client may have more than one IA assigned to it; for example, one for each of its interfaces. Each IA holds one type of address; for example, an identity association for temporary addresses (IA_TA) holds temporary addresses (see "identity association for temporary addresses") and identity association for prefix delegation (IA_PD) holds delegated prefixes. Throughout this document, "IA" is used to refer to an identity association without identifying the type of a lease in the IA. At the time of writing this document, there are 3 IA types defined: IA_NA, IA_TA and IA_PD. New IA types may be defined in the future.
IAID	Identity Association IDentifier: An identifier for an IA, chosen by the client. Each IA has an IAID, which is chosen to be unique among IAIDs for IAs of a specific type, belonging to that client.
IA_NA	Identity association for Non-temporary Addresses: An IA that carries assigned





addresses that are not temporary addresses (see "identity association for temporary addresses")

IA_TA	Identity Association for Temporary Addresses: An IA that carries temporary addresses (see [ <a href="#">RFC4941</a> ]).
IA_PD	Identity Association for Prefix Delegation: A collection of prefixes assigned to the requesting router. Each IA_PD has an associated IAID. A requesting router may have more than one IA_PD assigned to it; for example, one for each of its interfaces.
lease	It is an address assigned by the DHCP server to the client or a delegated prefix assigned by the delegating router to the requesting router. Leases are carried in the IA Address or IA Prefix options within the IA_NA/IA_TA and IA_PD options respectively.
message	A unit of data carried as the payload of a UDP datagram, exchanged among DHCP servers, relay agents and clients.
Reconfigure key	A key supplied to a client by a server used to provide security for Reconfigure messages.
requesting router	The router that acts as a DHCP client and is requesting prefix(es) to be assigned.
singleton option	An option that is allowed to appear only once. Most options are singletons.
relaying	A DHCP relay agent relays DHCP messages between DHCP participants.
transaction ID	An opaque value used to match responses with replies initiated either by a client or server.



## **5. Operational Models**

This section describes some of the current most common DHCP operational models. The described models are not mutually exclusive and are sometimes used together. For example, a device may start in stateful mode to obtain an address, and at a later time when an application is started, request additional parameters using stateless mode.

This document assumes that the DHCP servers and the client, communicating with the servers via specific interface, belong to a single provisioning domain.

### **5.1. Stateless DHCP**

Stateless DHCP [[RFC3736](#)] is used when DHCP is not used for obtaining a lease, but a node (DHCP client) desires one or more DHCP "other configuration" parameters, such as a list of DNS recursive name servers or DNS domain search lists [[RFC3646](#)]. Stateless may be used when a node initially boots or at any time the software on the node requires some missing or expired configuration information that is available via DHCP.

This is the simplest and most basic operation for DHCP and requires a client (and a server) to support only two messages - Information-request and Reply. Note that DHCP servers and relay agents typically also need to support the Relay-Forw and Relay-Reply messages to accommodate operation when clients and servers are not on the same link.

### **5.2. DHCP for Non-Temporary Address Assignment**

This model of operation was the original motivation for DHCP and is the "stateful address autoconfiguration protocol" for IPv6 [[RFC2462](#)]. It is appropriate for situations where stateless address autoconfiguration is not desired, because of network policy, additional requirements (such as updating the DNS with forward or reverse resource records), or client specific requirements (i.e., some prefixes are only available to some clients) which are not possible using stateless address autoconfiguration.

The model of operation for non-temporary address assignment is as follows. The server is provided with IPv6 prefixes from which it may allocate addresses to clients, as well as any related network topology information as to which prefixes are present on which links. A client requests a non-temporary address to be assigned by the server. The server allocates an address or addresses appropriate for



the link on which the client is connected. The server returns the allocated address or addresses to the client.

Each address has an associated preferred and valid lifetime, which constitutes an agreement about the length of time over which the client is allowed to use the address. A client can request an extension of the lifetimes on an address and is required to terminate the use of an address if the valid lifetime of the address expires.

Typically clients request other configuration parameters, such as the domain server addresses and search lists, when requesting addresses.

### **[5.3.](#) DHCP for Prefix Delegation**

The prefix delegation mechanism, originally described in [[RFC3633](#)], is another stateful mode of operation and intended for simple delegation of prefixes from a delegating router (DHCP server) to requesting routers (DHCP clients). It is appropriate for situations in which the delegating router does not have knowledge about the topology of the networks to which the requesting router is attached, and the delegating router does not require other information aside from the identity of the requesting router to choose a prefix for delegation. For example, these options would be used by a service provider to assign a prefix to a Customer Edge Router device acting as a router between the subscriber's internal network and the service provider's core network.

The design of this prefix delegation mechanism meets the requirements for prefix delegation in [[RFC3769](#)].

The model of operation for prefix delegation is as follows. A delegating router is provided IPv6 prefixes to be delegated to requesting routers. A requesting router requests prefix(es) from the delegating router, as described in [Section 19](#). The delegating router chooses prefix(es) for delegation, and responds with prefix(es) to the requesting router. The requesting router is then responsible for the delegated prefix(es). For example, the requesting router might assign a subnet from a delegated prefix to one of its interfaces, and begin sending router advertisements for the prefix on that link.

Each prefix has an associated valid and preferred lifetime, which constitutes an agreement about the length of time over which the requesting router is allowed to use the prefix. A requesting router can request an extension of the lifetimes on a delegated prefix and is required to terminate the use of a delegated prefix if the valid lifetime of the prefix expires.



The mechanism through which the delegating router selects prefix(es) for delegation is not specified in this document. Examples of ways in which the server might select prefix(es) for a client include: static assignment based on subscription to an ISP; dynamic assignment from a pool of available prefixes; selection based on an external authority such as a RADIUS server using the Framed-IPv6-Prefix option as described in [RFC3162].

This prefix delegation mechanism would be appropriate for use by an ISP to delegate a prefix to a subscriber, where the delegated prefix would possibly be subnetted and assigned to the links within the subscriber's network.

Figure 1 illustrates a network architecture in which prefix delegation could be used.

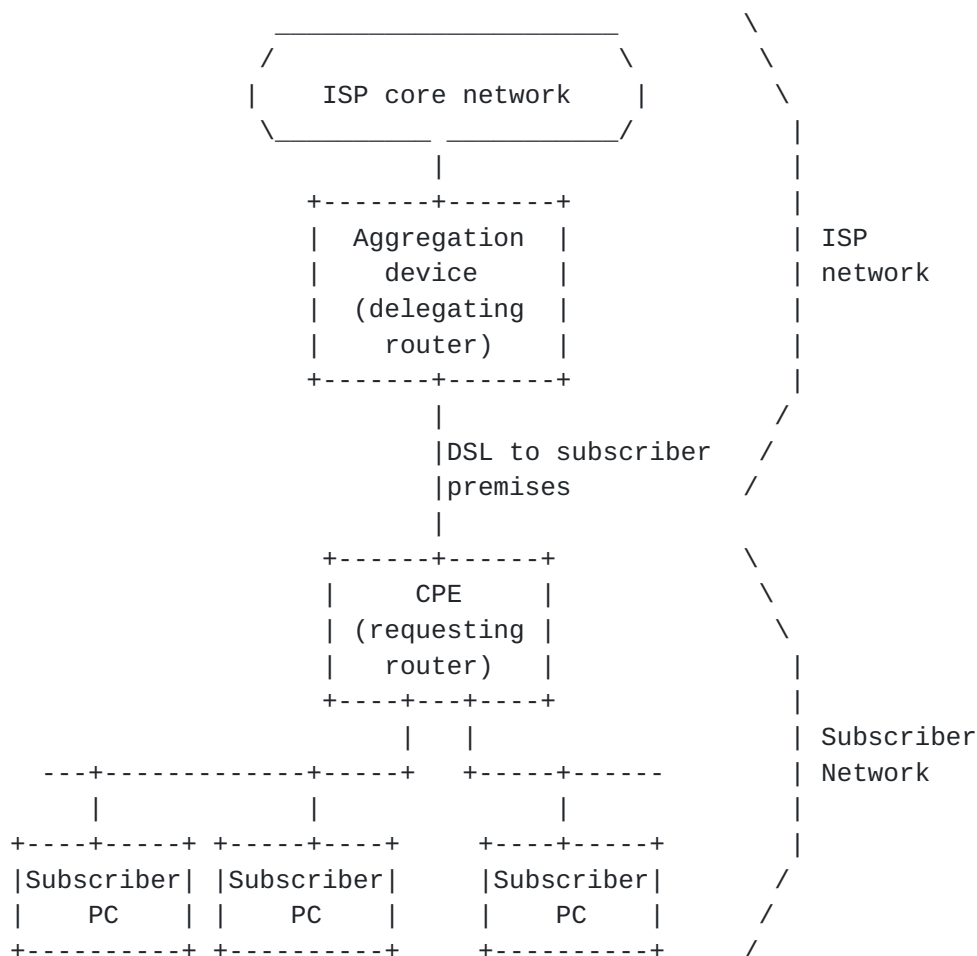


Figure 1: Prefix Delegation Network





In this example, the delegating router is configured with a set of prefixes to be used for assignment to customers at the time of each customer's first connection to the ISP service. The prefix delegation process begins when the requesting router requests configuration information through DHCP. The DHCP messages from the requesting router are received by the delegating router in the aggregation device. When the delegating router receives the request, it selects an available prefix or prefixes for delegation to the requesting router. The delegating router then returns the prefix or prefixes to the requesting router.

The requesting router subnets the delegated prefix and assigns the longer prefixes to links in the subscriber's network. In a typical scenario based on the network shown in Figure 1, the requesting router subnets a single delegated /48 prefix into /64 prefixes and assigns one /64 prefix to each of the links in the subscriber network.

The prefix delegation options can be used in conjunction with other DHCP options carrying other configuration information to the requesting router. The requesting router may, in turn, provide DHCP service to hosts attached to the internal network. For example, the requesting router may obtain the addresses of DNS and NTP servers from the ISP delegating router, and then pass that configuration information on to the subscriber hosts through a DHCP server in the requesting router.

If the requesting router assigns a delegated prefix to a link to which the router is attached, and begins to send router advertisements for the prefix on the link, the requesting router **MUST** set the valid lifetime in those advertisements to be no later than the valid lifetime specified in the IA\_PD Prefix option. A requesting router **MAY** use the preferred lifetime specified in the IA\_PD Prefix option.

#### **5.4. DHCP for Customer Edge Routers**

The DHCP requirements and network architecture for Customer Edge Routers are described in [[RFC7084](#)]. This model of operation combines address assignment (see [Section 5.2](#)) and prefix delegation (see [Section 5.3](#)). In general, this model assumes that a single set of transactions between the client and server will assign or extend the client's non-temporary addresses and delegated prefixes.



### **[5.5.](#) DHCP for Temporary Addresses**

Temporary addresses were originally introduced to avoid privacy concerns with stateless address autoconfiguration, which based 64-bits of the address on the EUI-64 (see [[RFC3041](#)] and [[RFC4941](#)]). They were added to DHCP to provide complementary support when stateful address assignment is used.

Temporary address assignment works mostly like non-temporary address assignment (see [Section 5.2](#)), however these addresses are generally intended to be used for a short period of time and not to have their lifetimes extended, though they can be if required.

## **[6.](#) DHCP Constants**

This section describes various program and networking constants used by DHCP.

### **[6.1.](#) Multicast Addresses**

DHCP makes use of the following multicast addresses:

All\_DHCP\_Relay\_Agents\_and\_Servers (FF02::1:2) A link-scoped multicast address used by a client to communicate with neighboring (i.e., on-link) relay agents and servers. All servers and relay agents are members of this multicast group.

All\_DHCP\_Servers (FF05::1:3) A site-scoped multicast address used by a relay agent to communicate with servers, either because the relay agent wants to send messages to all servers or because it does not know the unicast addresses of the servers. Note that in order for a relay agent to use this address, it must have an address of sufficient scope to be reachable by the servers. All servers within the site are members of this multicast group.

### **[6.2.](#) UDP Ports**

Clients listen for DHCP messages on UDP port 546. Servers and relay agents listen for DHCP messages on UDP port 547.

### **[6.3.](#) DHCP Message Types**

DHCP defines the following message types. More detail on these message types can be found in [Section 7](#) and [Section 8](#). Message types



not listed here are reserved for future use. The numeric encoding for each message type is shown in parentheses.

- SOLICIT (1)      A client sends a Solicit message to locate servers.
- ADVERTISE (2)    A server sends an Advertise message to indicate that it is available for DHCP service, in response to a Solicit message received from a client.
- REQUEST (3)     A client sends a Request message to request configuration parameters, including IP addresses, from a specific server.
- CONFIRM (4)     A client sends a Confirm message to any available server to determine whether the addresses it was assigned are still appropriate to the link to which the client is connected.
- RENEW (5)       A client sends a Renew message to the server that originally provided the client's addresses and configuration parameters to extend the lifetimes on the addresses assigned to the client and to update other configuration parameters.
- REBIND (6)      A client sends a Rebind message to any available server to extend the lifetimes on the addresses assigned to the client and to update other configuration parameters; this message is sent after a client receives no response to a Renew message.
- REPLY (7)       A server sends a Reply message containing assigned addresses and configuration parameters in response to a Solicit, Request, Renew, Rebind message received from a client. A server sends a Reply message containing configuration parameters in response to an Information-request message. A server sends a Reply message in response to a Confirm message confirming or denying that the addresses assigned to the client are appropriate to the link to which the client is connected. A server sends a Reply message to acknowledge receipt of a Release or Decline message.
- RELEASE (8)     A client sends a Release message to the server that assigned addresses to the client to indicate that the client will no longer use one or more of the assigned addresses.



- DECLINE (9) A client sends a Decline message to a server to indicate that the client has determined that one or more addresses assigned by the server are already in use on the link to which the client is connected.
- RECONFIGURE (10) A server sends a Reconfigure message to a client to inform the client that the server has new or updated configuration parameters, and that the client is to initiate a Renew/Reply or Information-request/Reply transaction with the server in order to receive the updated information.
- INFORMATION-REQUEST (11) A client sends an Information-request message to a server to request configuration parameters without the assignment of any IP addresses to the client.
- RELAY-FORW (12) A relay agent sends a Relay-forward message to relay messages to servers, either directly or through another relay agent. The received message, either a client message or a Relay-forward message from another relay agent, is encapsulated in an option in the Relay-forward message.
- RELAY-REPL (13) A server sends a Relay-reply message to a relay agent containing a message that the relay agent delivers to a client. The Relay-reply message may be relayed by other relay agents for delivery to the destination relay agent.

The server encapsulates the client message as an option in the Relay-reply message, which the relay agent extracts and relays to the client.

#### **[6.4.](#) Status Codes**

DHCPv6 uses status codes to communicate the success or failure of operations requested in messages from clients and servers, and to provide additional information about the specific cause of the failure of a message. The specific status codes are defined in [Section 23.12](#).

If the Status Code option does not appear in a message in which the option could appear, the status of the message is assumed to be Success.





### 6.5. Transmission and Retransmission Parameters

This section presents a table of values used to describe the message transmission behavior of clients and servers.

Parameter	Default	Description
SOL_MAX_DELAY	1 sec	Max delay of first Solicit
SOL_TIMEOUT	1 sec	Initial Solicit timeout
SOL_MAX_RT	3600 secs	Max Solicit timeout value
REQ_TIMEOUT	1 sec	Initial Request timeout
REQ_MAX_RT	30 secs	Max Request timeout value
REQ_MAX_RC	10	Max Request retry attempts
CNF_MAX_DELAY	1 sec	Max delay of first Confirm
CNF_TIMEOUT	1 sec	Initial Confirm timeout
CNF_MAX_RT	4 secs	Max Confirm timeout
CNF_MAX_RD	10 secs	Max Confirm duration
REN_TIMEOUT	10 secs	Initial Renew timeout
REN_MAX_RT	600 secs	Max Renew timeout value
REB_TIMEOUT	10 secs	Initial Rebind timeout
REB_MAX_RT	600 secs	Max Rebind timeout value
INF_MAX_DELAY	1 sec	Max delay of first Information-request
INF_TIMEOUT	1 sec	Initial Information-request timeout
INF_MAX_RT	3600 secs	Max Information-request timeout value
REL_TIMEOUT	1 sec	Initial Release timeout
REL_MAX_RC	4	MAX Release retry attempts
DEC_TIMEOUT	1 sec	Initial Decline timeout
DEC_MAX_RC	4	Max Decline retry attempts
REC_TIMEOUT	2 secs	Initial Reconfigure timeout
REC_MAX_RC	8	Max Reconfigure attempts
HOP_COUNT_LIMIT	32	Max hop count in a Relay-forward message

### 6.6. Representation of time values and "Infinity" as a time value

All time values for lifetimes, T1 and T2 are unsigned integers. The value 0xffffffff is taken to mean "infinity" when used as a lifetime (as in [[RFC4861](#)]) or a value for T1 or T2.

## 7. Client/Server Message Formats

All DHCP messages sent between clients and servers share an identical fixed format header and a variable format area for options.



All values in the message header and in options are in network byte order.

Options are stored serially in the options field, with no padding between the options. Options are byte-aligned but are not aligned in any other way such as on 2 or 4 byte boundaries.

The following diagram illustrates the format of DHCP messages sent between clients and servers:

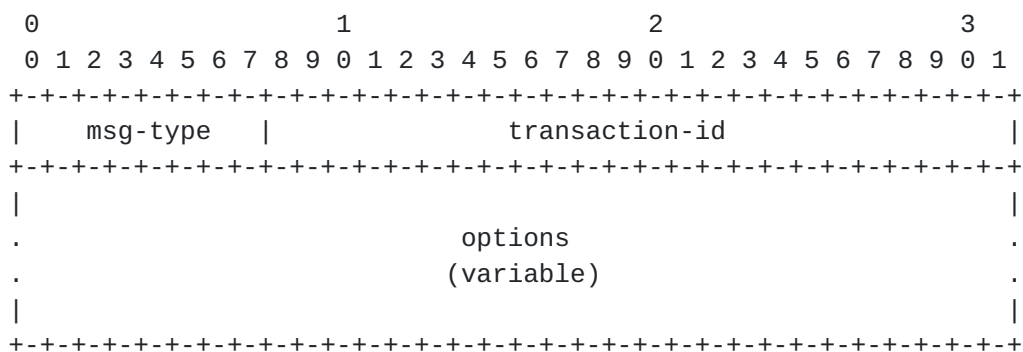


Figure 2: Client/Server message format

msg-type	Identifies the DHCP message type; the available message types are listed in <a href="#">Section 6.3</a> .
transaction-id	The transaction ID for this message exchange.
options	Options carried in this message; options are described in <a href="#">Section 23</a> .

## 8. Relay Agent/Server Message Formats

Relay agents exchange messages with servers to relay messages between clients and servers that are not connected to the same link.

All values in the message header and in options are in network byte order.

Options are stored serially in the options field, with no padding between the options. Options are byte-aligned but are not aligned in any other way such as on 2 or 4 byte boundaries.

There are two relay agent messages, which share the following format:



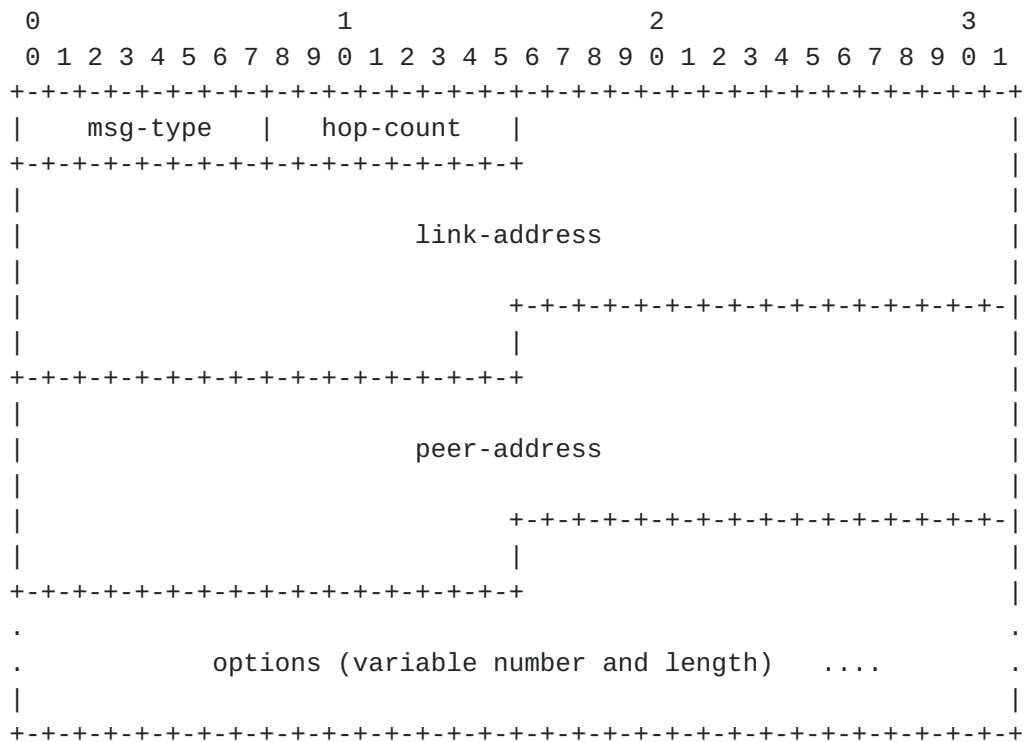


Figure 3: Relay Agent/Server message format

The following sections describe the use of the Relay Agent message header.

**8.1. Relay-forward Message**

The following table defines the use of message fields in a Relay-forward message.

msg-type	RELAY-FORW
hop-count	Number of relay agents that have relayed this message.
link-address	An address that will be used by the server to identify the link on which the client is located. This is typically global, site-scoped or ULA [ <a href="#">RFC4193</a> ], but see discussion in <a href="#">Section 21.1.1</a> .
peer-address	The address of the client or relay agent from which the message to be relayed was received.



options                    MUST include a "Relay Message option" (see [Section 23.10](#)); MAY include other options added by the relay agent.

## **8.2. Relay-reply Message**

The following table defines the use of message fields in a Relay-reply message.

msg-type	RELAY-REPL
hop-count	Copied from the Relay-forward message
link-address	Copied from the Relay-forward message
peer-address	Copied from the Relay-forward message
options	MUST include a "Relay Message option"; see <a href="#">Section 23.10</a> ; MAY include other options

## **9. Representation and Use of Domain Names**

So that domain names may be encoded uniformly, a domain name or a list of domain names is encoded using the technique described in [section 3.1 of \[RFC1035\]](#). A domain name, or list of domain names, in DHCP MUST NOT be stored in compressed form, as described in [section 4.1.4 of \[RFC1035\]](#).

## **10. DHCP Unique Identifier (DUID)**

Each DHCP client and server has a DUID. DHCP servers use DUIDs to identify clients for the selection of configuration parameters and in the association of IAs with clients. DHCP clients use DUIDs to identify a server in messages where a server needs to be identified. See [Section 23.2](#) and [Section 23.3](#) for the representation of a DUID in a DHCP message.

Clients and servers MUST treat DUIDs as opaque values and MUST only compare DUIDs for equality. Clients and servers MUST NOT in any other way interpret DUIDs. Clients and servers MUST NOT restrict DUIDs to the types defined in this document, as additional DUID types may be defined in the future.

The DUID is carried in an option because it may be variable length and because it is not required in all DHCP messages. The DUID is designed to be unique across all DHCP clients and servers, and stable for any specific client or server - that is, the DUID used by a client or server SHOULD NOT change over time if at all possible; for





example, a device's DUID should not change as a result of a change in the device's network hardware.

The motivation for having more than one type of DUID is that the DUID must be globally unique, and must also be easy to generate. The sort of globally-unique identifier that is easy to generate for any given device can differ quite widely. Also, some devices may not contain any persistent storage. Retaining a generated DUID in such a device is not possible, so the DUID scheme must accommodate such devices.

### [10.1.](#) DUID Contents

A DUID consists of a two-octet type code represented in network byte order, followed by a variable number of octets that make up the actual identifier. The length of the DUID (not including the type code) is at least 1 octet and at most 128 octets. The following types are currently defined:

+-----+-----+-----+-----+-----+-----+	
Type	Description
+-----+-----+-----+-----+-----+-----+	
1	Link-layer address plus time
2	Vendor-assigned unique ID based on Enterprise Number
3	Link-layer address
4	Universally Unique IDentifier (UUID) - see [ <a href="#">RFC6355</a> ]
+-----+-----+-----+-----+-----+-----+	

Formats for the variable field of the DUID for the first 3 of the above types are shown below. The fourth type, DUID-UUID [[RFC6355](#)], can be used in situations where there is a UUID stored in a device's firmware settings.

### [10.2.](#) DUID Based on Link-layer Address Plus Time, DUID-LLT

This type of DUID consists of a two octet type field containing the value 1, a two octet hardware type code, four octets containing a time value, followed by link-layer address of any one network interface that is connected to the DHCP device at the time that the DUID is generated. The time value is the time that the DUID is generated represented in seconds since midnight (UTC), January 1, 2000, modulo  $2^{32}$ . The hardware type MUST be a valid hardware type assigned by the IANA as described in [[RFC0826](#)]. Both the time and the hardware type are stored in network byte order. The link-layer address is stored in canonical form, as described in [[RFC2464](#)].

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



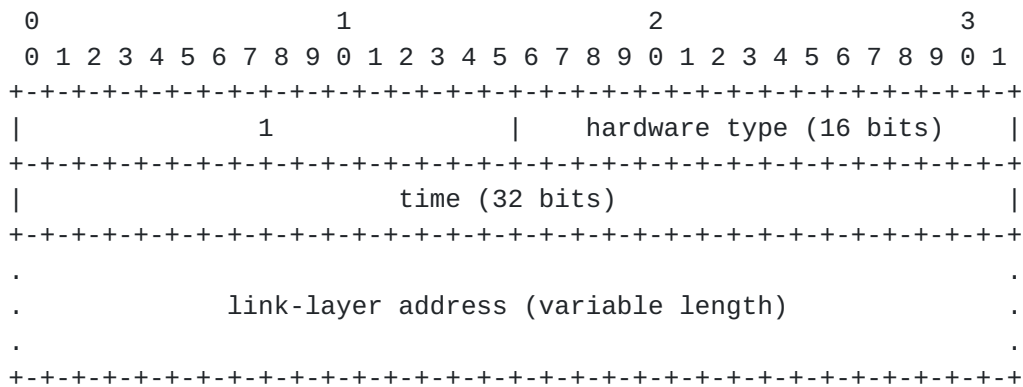


Figure 4: DUID-LLT format

The choice of network interface can be completely arbitrary, as long as that interface provides a globally unique link-layer address for the link type, and the same DUID-LLT SHOULD be used in configuring all network interfaces connected to the device, regardless of which interface's link-layer address was used to generate the DUID-LLT.

Clients and servers using this type of DUID MUST store the DUID-LLT in stable storage, and MUST continue to use this DUID-LLT even if the network interface used to generate the DUID-LLT is removed. Clients and servers that do not have any stable storage MUST NOT use this type of DUID.

Clients and servers that use this DUID SHOULD attempt to configure the time prior to generating the DUID, if that is possible, and MUST use some sort of time source (for example, a real-time clock) in generating the DUID, even if that time source could not be configured prior to generating the DUID. The use of a time source makes it unlikely that two identical DUID-LLTs will be generated if the network interface is removed from the client and another client then uses the same network interface to generate a DUID-LLT. A collision between two DUID-LLTs is very unlikely even if the clocks have not been configured prior to generating the DUID.

This method of DUID generation is recommended for all general purpose computing devices such as desktop computers and laptop computers, and also for devices such as printers, routers, and so on, that contain some form of writable non-volatile storage.

Despite our best efforts, it is possible that this algorithm for generating a DUID could result in a client identifier collision. A DHCP client that generates a DUID-LLT using this mechanism MUST provide an administrative interface that replaces the existing DUID with a newly-generated DUID-LLT.



### 10.3. DUID Assigned by Vendor Based on Enterprise Number, DUID-EN

This form of DUID is assigned by the vendor to the device. It consists of the vendor's registered Private Enterprise Number as maintained by IANA [[IANA-PEN](#)] followed by a unique identifier assigned by the vendor. The following diagram summarizes the structure of a DUID-EN:

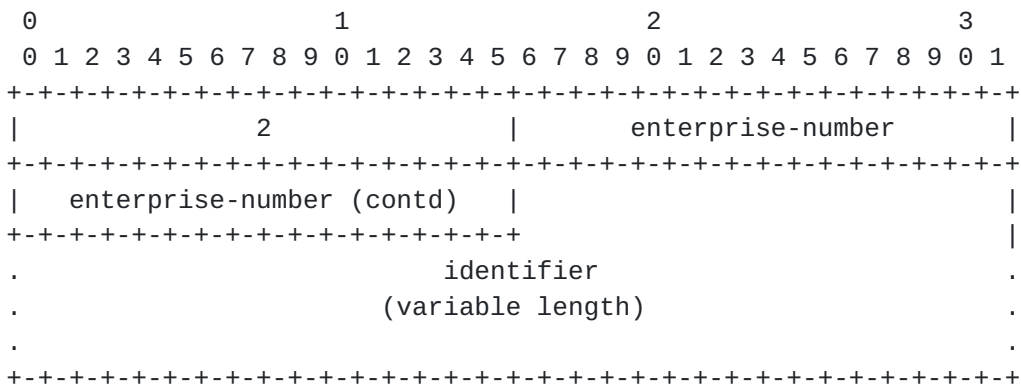


Figure 5: DUID-EN format

The source of the identifier is left up to the vendor defining it, but each identifier part of each DUID-EN MUST be unique to the device that is using it, and MUST be assigned to the device no later than at the first usage and stored in some form of non-volatile storage. This typically means being assigned during manufacture process in case of physical devices or when the image is created or booted for the first time in case of virtual machines. The generated DUID SHOULD be recorded in non-erasable storage. The enterprise-number is the vendor's registered Private Enterprise Number as maintained by IANA [[IANA-PEN](#)]. The enterprise-number is stored as an unsigned 32 bit number.

An example DUID of this type might look like this:

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | 2 | 0 | 0 | 0 | 9 | 12 | 192 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 132 | 211 | 3 | 0 | 9 | 18 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 6: DUID-EN example



This example includes the two-octet type of 2, the Enterprise Number (9), followed by eight octets of identifier data (0x0CC084D303000912).

#### 10.4. DUID Based on Link-layer Address, DUID-LL

This type of DUID consists of two octets containing the DUID type 3, a two octet network hardware type code, followed by the link-layer address of any one network interface that is permanently connected to the client or server device. For example, a host that has a network interface implemented in a chip that is unlikely to be removed and used elsewhere could use a DUID-LL. The hardware type **MUST** be a valid hardware type assigned by the IANA, as described in [RFC0826]. The hardware type is stored in network byte order. The link-layer address is stored in canonical form, as described in [RFC2464]. The following diagram illustrates the format of a DUID-LL:

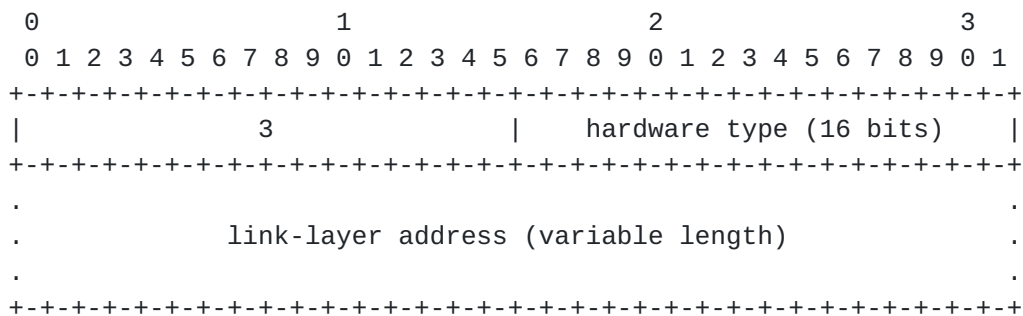


Figure 7: DUID-LL format

The choice of network interface can be completely arbitrary, as long as that interface provides a unique link-layer address and is permanently attached to the device on which the DUID-LL is being generated. The same DUID-LL **SHOULD** be used in configuring all network interfaces connected to the device, regardless of which interface's link-layer address was used to generate the DUID.

DUID-LL is recommended for devices that have a permanently-connected network interface with a link-layer address, and do not have nonvolatile, writable stable storage. DUID-LL **MUST NOT** be used by DHCP clients or servers that cannot tell whether or not a network interface is permanently attached to the device on which the DHCP client is running.





## **11. Identity Association**

An "identity-association" (IA) is a construct through which a server and a client can identify, group, and manage a set of related IPv6 addresses or delegated prefixes. Each IA consists of an IAID and associated configuration information.

The IAID uniquely identifies the IA and must be chosen to be unique among the IAIDs for that IA type on the client. The IAID is chosen by the client. For any given use of an IA by the client, the IAID for that IA MUST be consistent across restarts of the DHCP client. The client may maintain consistency either by storing the IAID in non-volatile storage or by using an algorithm that will consistently produce the same IAID as long as the configuration of the client has not changed. There may be no way for a client to maintain consistency of the IAIDs if it does not have non-volatile storage and the client's hardware configuration changes. If the client uses only one IAID, it can use a well-known value, e.g., zero.

### **11.1. Identity Associations for Address Assignment**

A client must associate at least one distinct IA with each of its network interfaces for which it is to request the assignment of IPv6 addresses from a DHCP server. The client uses the IAs assigned to an interface to obtain configuration information from a server for that interface. Each IA must be associated with exactly one interface.

The configuration information in an IA consists of one or more IPv6 addresses along with the times T1 and T2 for the IA. See [Section 23.4](#) for the representation of an IA in a DHCP message.

Each address in an IA has a preferred lifetime and a valid lifetime, as defined in [[RFC4862](#)]. The lifetimes are transmitted from the DHCP server to the client in the IA option. The lifetimes apply to the use of IPv6 addresses, as described in [section 5.5.4 of \[RFC4862\]](#).

### **11.2. Identity Associations for Prefix Delegation**

An IA\_PD is different from an IA for address assignment, in that it does not need to be associated with exactly one interface. One IA\_PD can be associated with the requesting router, with a set of interfaces or with exactly one interface. A requesting router must create at least one distinct IA\_PD. It may associate a distinct IA\_PD with each of its downstream network interfaces and use that IA\_PD to obtain a prefix for that interface from the delegating router.



The configuration information in an IA\_PD consists of one or more IPv6 prefixes along with the times T1 and T2 for the IA\_PD. See [Section 23.21](#) for the representation of an IA\_PD in a DHCP message.

## **12. Selecting Addresses for Assignment to an IA**

A server selects addresses to be assigned to an IA according to the address assignment policies determined by the server administrator and the specific information the server determines about the client from some combination of the following sources:

- The link to which the client is attached. The server determines the link as follows:
  - \* If the server receives the message directly from the client and the source address in the IP datagram in which the message was received is a link-local address, then the client is on the same link to which the interface over which the message was received is attached.
  - \* If the server receives the message from a forwarding relay agent, then the client is on the same link as the one to which the interface, identified by the link-address field in the message from the relay agent, is attached. According to [\[RFC6221\]](#), the server MUST ignore any link-address field whose value is zero. The link address field refers to the link-address field of the Relay-Forward message, and the link-address fields in any Relay-Forward messages that may be nested within the Relay-Forward message.
  - \* If the server receives the message directly from the client and the source address in the IP datagram in which the message was received is not a link-local address, then the client is on the link identified by the source address in the IP datagram (note that this situation can occur only if the server has enabled the use of unicast message delivery by the client and the client has sent a message for which unicast delivery is allowed).
- The DUID supplied by the client.
- Other information in options supplied by the client, e.g. IA Address options that include the client's requests for specific addresses.
- Other information in options supplied by the relay agent.



Any address assigned by a server that is based on an EUI-64 identifier MUST include an interface identifier with the "u" (universal/local) and "g" (individual/group) bits of the interface identifier set appropriately, as indicated in [section 2.5.1 of \[RFC4291\]](#).

A server MUST NOT assign an address that is otherwise reserved for some other purpose. For example, a server MUST NOT assign reserved anycast addresses, as defined in [\[RFC2526\]](#), from any subnet.

### **[13.](#) Management of Temporary Addresses**

A client may request the assignment of temporary addresses (see [\[RFC4941\]](#) for the definition of temporary addresses). DHCPv6 handling of address assignment is no different for temporary addresses.

Clients ask for temporary addresses and servers assign them. Temporary addresses are carried in the Identity Association for Temporary Addresses (IA\_TA) option (see [Section 23.5](#)). Each IA\_TA option contains at most one temporary address for each of the prefixes on the link to which the client is attached.

The lifetime of the assigned temporary address is set in the IA Address Option (see [Section 23.6](#)) within the IA\_TA option. It is RECOMMENDED to set short lifetimes, typically shorter than TEMP\_VALID\_LIFETIME and TEMP\_PREFERRED\_LIFETIME (see [Section 5, \[RFC4941\]](#)).

The IAID number space for the IA\_TA option IAID number space is separate from the IA\_NA option IAID number space.

A DHCPv6 server implementation MAY generate temporary addresses referring to the algorithm defined in [Section 3.2.1, \[RFC4941\]](#), with additional condition that the new address is not duplicated with any assigned addresses.

The server MAY update the DNS for a temporary address, as described in [section 4 of \[RFC4941\]](#).

On the clients, by default, temporary addresses are preferred in source address selection, according to Rule 7, [\[RFC6724\]](#). However, this policy is overridable.

One of the most important properties of temporary address is unlinkability of different actions over time. So, it is NOT RECOMMENDED for a client to renew expired temporary addresses, though DHCPv6 provides such possibility (see [Section 23.5](#)).



## **14. Transmission of Messages by a Client**

Unless otherwise specified in this document, or in a document that describes how IPv6 is carried over a specific type of link (for link types that do not support multicast), a client sends DHCP messages to the All\_DHCP\_Relay\_Agents\_and\_Servers.

A client uses multicast to reach all servers or an individual server. An individual server is indicated by specifying that server's DUID in a Server Identifier option (see [Section 23.3](#)) in the client's message (all servers will receive this message but only the indicated server will respond). All servers are indicated by not supplying this option.

A client may send some messages directly to a server using unicast, as described in [Section 23.12](#).

### **14.1. Rate Limiting**

In order to avoid prolonged message bursts that may be caused by possible logic loops, a DHCPv6 client MUST limit the rate of DHCPv6 messages it transmits. One example is that a client obtains an address, but does not like the response; it reverts back to Solicit procedure, discovers the same (sole) server, requests an address and gets the same address as before (the server still has the lease that was requested just previously). This loops can repeat infinitely if there is not a quit/stop mechanism. Therefore, a client must not initiate transmissions too frequently.

A recommended method for implementing the rate limiting function is a token bucket, limiting the average rate of transmission to a certain number in a certain time. This method of bounding burstiness also guarantees that the long-term transmission rate will not exceed.

TRT      Transmission Rate Limit

The Transmission Rate Limit parameter (TRT) SHOULD be configurable. A possible default could be 20 packets in 20 seconds.

For a device that has multiple interfaces, the limit MUST be enforced on a per interface basis.

Rate limiting of forwarded DHCPv6 messages and server-side messages are out of scope of this specification.





#### **14.2. Choosing transmission time when T1,T2 is 0**

In certain cases, T1 and/or T2 timers may be set to zero. Currently there are three such cases: 1. a client received IA\_NA option with zeroed values; 2. a client received IA\_PD option with zeroed values; 3. a client received IA\_TA option (which does not contain T1 or T2 fields). Additional cases may appear in the future. This is an indication that the transmission times are left at client's discretion. They are not completely discretionary, though.

When T1 and/or T2 timers are set to zero, client MUST choose transmission time to avoid packet storms. In particular, it MUST NOT transmit immediately. If the client received multiple IA containers, it SHOULD pick renew and/or rebind transmission time so all IA containers are received in one exchange, if possible. Client MUST choose the transmission times to not violate rate limiting restrictions, defined in [Section 14.1](#).

#### **15. Reliability of Client Initiated Message Exchanges**

DHCP clients are responsible for reliable delivery of messages in the client-initiated message exchanges described in [Section 18](#) and [Section 19](#). If a DHCP client fails to receive an expected response from a server, the client must retransmit its message. This section describes the retransmission strategy to be used by clients in client-initiated message exchanges.

Note that the procedure described in this section is slightly modified when used with the Solicit message. The modified procedure is described in [Section 18.1.2](#).

The client begins the message exchange by transmitting a message to the server. The message exchange terminates when either the client successfully receives the appropriate response or responses from a server or servers, or when the message exchange is considered to have failed according to the retransmission mechanism described below.

The client retransmission behavior is controlled and described by the following variables:

RT	Retransmission timeout
IRT	Initial retransmission time
MRC	Maximum retransmission count
MRT	Maximum retransmission time



MRD      Maximum retransmission duration

RAND     Randomization factor

With each message transmission or retransmission, the client sets RT according to the rules given below. If RT expires before the message exchange terminates, the client recomputes RT and retransmits the message.

Each of the computations of a new RT include a randomization factor (RAND), which is a random number chosen with a uniform distribution between -0.1 and +0.1. The randomization factor is included to minimize synchronization of messages transmitted by DHCP clients.

The algorithm for choosing a random number does not need to be cryptographically sound. The algorithm SHOULD produce a different sequence of random numbers from each invocation of the DHCP client.

RT for the first message transmission is based on IRT:

$$RT = IRT + RAND * IRT$$

RT for each subsequent message transmission is based on the previous value of RT:

$$RT = 2 * RT_{prev} + RAND * RT_{prev}$$

MRT specifies an upper bound on the value of RT (disregarding the randomization added by the use of RAND). If MRT has a value of 0, there is no upper limit on the value of RT. Otherwise:

$$\begin{aligned} &\text{if } (RT > MRT) \\ &\quad RT = MRT + RAND * MRT \end{aligned}$$

MRC specifies an upper bound on the number of times a client may retransmit a message. Unless MRC is zero, the message exchange fails once the client has transmitted the message MRC times.

MRD specifies an upper bound on the length of time a client may retransmit a message. Unless MRD is zero, the message exchange fails once MRD seconds have elapsed since the client first transmitted the message.

If both MRC and MRD are non-zero, the message exchange fails whenever either of the conditions specified in the previous two paragraphs are met.



If both MRC and MRD are zero, the client continues to transmit the message until it receives a response.

A client is not expected to listen for a response during the entire period between transmission of Solicit or Information-request messages.

## **16. Message Validation**

Clients and servers might get messages that contain options not allowed to appear in the received message. For example, an IA option is not allowed to appear in an Information-request message. Clients and servers MAY choose either to extract information from such a message if the information is of use to the recipient, or to ignore such message completely and just drop it.

If a server receives a message that contains options it should not contain (such as an Information-request message with an IA option), is missing options that it should contain, or is otherwise not valid, it MAY send a Reply (or Advertise as appropriate) with a Server Identifier option, a Client Identifier option if one was included in the message and a Status Code option with status UnSpecFail.

Clients, relay agents and servers MUST NOT discard messages that contain unknown options (or instances of vendor options with unknown enterprise-numbers). These should be ignored as if they weren't present.

A server MUST discard any Solicit, Confirm, Rebind or Information-request messages it receives with a unicast destination address.

A client or server MUST silently discard any received DHCPv6 messages with an unknown message type.

### **16.1. Use of Transaction IDs**

The "transaction-id" field holds a value used by clients and servers to synchronize server responses to client messages. A client SHOULD generate a random number that cannot easily be guessed or predicted to use as the transaction ID for each new message it sends. Note that if a client generates easily predictable transaction identifiers, it may become more vulnerable to certain kinds of attacks from off-path intruders. A client MUST leave the transaction ID unchanged in retransmissions of a message.



### **[16.2.](#) Solicit Message**

Clients MUST discard any received Solicit messages.

Servers MUST discard any Solicit messages that do not include a Client Identifier option or that do include a Server Identifier option.

### **[16.3.](#) Advertise Message**

Clients MUST discard any received Advertise message that meets any of the following conditions:

- the message does not include a Server Identifier option.
- the message does not include a Client Identifier option.
- the contents of the Client Identifier option does not match the client's DUID.
- the "transaction-id" field value does not match the value the client used in its Solicit message.

Servers and relay agents MUST discard any received Advertise messages.

### **[16.4.](#) Request Message**

Clients MUST discard any received Request messages.

Servers MUST discard any received Request message that meets any of the following conditions:

- the message does not include a Server Identifier option.
- the contents of the Server Identifier option do not match the server's DUID.
- the message does not include a Client Identifier option.

### **[16.5.](#) Confirm Message**

Clients MUST discard any received Confirm messages.

Servers MUST discard any received Confirm messages that do not include a Client Identifier option or that do include a Server Identifier option.





### **[16.6.](#) Renew Message**

Clients MUST discard any received Renew messages.

Servers MUST discard any received Renew message that meets any of the following conditions:

- the message does not include a Server Identifier option.
- the contents of the Server Identifier option does not match the server's identifier.
- the message does not include a Client Identifier option.

### **[16.7.](#) Rebind Message**

Clients MUST discard any received Rebind messages.

Servers MUST discard any received Rebind messages that do not include a Client Identifier option or that do include a Server Identifier option.

### **[16.8.](#) Decline Messages**

Clients MUST discard any received Decline messages.

Servers MUST discard any received Decline message that meets any of the following conditions:

- the message does not include a Server Identifier option.
- the contents of the Server Identifier option does not match the server's identifier.
- the message does not include a Client Identifier option.

### **[16.9.](#) Release Message**

Clients MUST discard any received Release messages.

Servers MUST discard any received Release message that meets any of the following conditions:

- the message does not include a Server Identifier option.
- the contents of the Server Identifier option does not match the server's identifier.



- the message does not include a Client Identifier option.

#### **16.10. Reply Message**

Clients MUST discard any received Reply message that meets any of the following conditions:

- the message does not include a Server Identifier option.
- the "transaction-id" field in the message does not match the value used in the original message.

If the client included a Client Identifier option in the original message, the Reply message MUST include a Client Identifier option and the contents of the Client Identifier option MUST match the DUID of the client; OR, if the client did not include a Client Identifier option in the original message, the Reply message MUST NOT include a Client Identifier option.

Servers and relay agents MUST discard any received Reply messages.

#### **16.11. Reconfigure Message**

Servers and relay agents MUST discard any received Reconfigure messages.

Clients MUST discard any Reconfigure message that meets any of the following conditions:

- the message was not unicast to the client.
- the message does not include a Server Identifier option.
- the message does not include a Client Identifier option that contains the client's DUID.
- the message does not contain a Reconfigure Message option.
- the Reconfigure Message option msg-type is not a valid value.
- the message includes any IA options and the msg-type in the Reconfigure Message option is INFORMATION-REQUEST.
- the message does not include DHCP authentication:
  - \* the message does not contain an authentication option.



- \* the message does not pass the authentication validation performed by the client.

#### **16.12. Information-request Message**

Clients MUST discard any received Information-request messages.

Servers MUST discard any received Information-request message that meets any of the following conditions:

- The message includes a Server Identifier option and the DUID in the option does not match the server's DUID.
- The message includes an IA option.

#### **16.13. Relay-forward Message**

Clients MUST discard any received Relay-forward messages.

#### **16.14. Relay-reply Message**

Clients and servers MUST discard any received Relay-reply messages.

### **17. Client Source Address and Interface Selection**

Client's behavior is different depending on the purpose of the configuration.

#### **17.1. Address Assignment**

When a client sends a DHCP message to the All\_DHCP\_Relay\_Agents\_and\_Servers address, it SHOULD send the message through the interface for which configuration information is being requested. However, the client MAY send the message through another interface if the interface is a logical interface without direct link attachment or the client is certain that two interfaces are attached to the same link.

When a client sends a DHCP message directly to a server using unicast (after receiving the Server Unicast option from that server), the source address in the header of the IPv6 datagram MUST be an address assigned to the interface for which the client is interested in obtaining configuration and which is suitable for use by the server in responding to the client.



### **[17.2.](#) Prefix Delegation**

Delegated prefixes are not associated with a particular interface in the same way as addresses are for address assignment, and mentioned above.

When a client (acting as requesting router) sends a DHCP message for the purpose of prefix delegation, it **SHOULD** be sent on the interface associated with the upstream router (ISP network). The upstream interface is typically determined by configuration. This rule applies even in the case where a separate IA\_PD is used for each downstream interface.

When a requesting router sends a DHCP message directly to a delegating router using unicast (after receiving the Server Unicast option from that delegating router), the source address **SHOULD** be an address from the upstream interface and which is suitable for use by the delegating router in responding to the requesting router.

## **[18.](#) DHCP Server Solicitation**

This section describes how a client locates servers that will assign addresses and delegated prefixes to IAs belonging to the client.

The client is responsible for creating IAs and requesting that a server assign IPv6 addresses and/or delegated prefixes to the IAs. The client first creates the IAs and assigns IAIDs to them. The client then transmits a Solicit message containing the IA options describing the IAs. The client **MUST NOT** be using any of the addresses or delegated prefixes for which it tries to obtain the bindings by sending the Solicit message. In particular, if the client had some valid bindings and has chosen to start the server solicitation process to obtain the bindings from a different server, the client **MUST** stop using the addresses and delegated prefixes for the bindings it had obtained from the previous server, and which it is now trying to obtain from a new server.

Servers that can assign addresses or delegated prefixes to the IAs respond to the client with an Advertise message. The client then initiates a configuration exchange as described in [Section 19](#).

If the client will accept a Reply message with committed leases assignments and other resources in response to the Solicit message, the client includes a Rapid Commit option (see [Section 23.14](#)) in the Solicit message.





### **[18.1.](#) Client Behavior**

A client uses the Solicit message to discover DHCP servers configured to assign leases or return other configuration parameters on the link to which the client is attached.

#### **[18.1.1.](#) Creation of Solicit Messages**

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

The client MUST include a Client Identifier option to identify itself to the server. The client includes IA options for any IAs to which it wants the server to assign leases.

The client MAY include addresses in the IA\_NA and IA\_TA options as hints to the server about the addresses for which the client has a preference.

The client MAY include values in the IA Prefix option encapsulated within IA\_PD option as hints for the delegated prefix and/or prefix length for which the client has a preference.

The client MUST NOT include any other options in the Solicit message, except as specifically allowed in the definition of individual options.

The client uses IA\_NA options to request the assignment of non-temporary addresses, IA\_TA options to request the assignment of temporary addresses and IA\_PD options to request prefix delegation. Either IA\_NA, IA\_TA or IA\_PD options, or a combination of all, can be included in DHCP messages. In addition, multiple instances of any IA option type can be included.

The client MUST include an Option Request option (see [Section 23.7](#)) to request the SOL\_MAX\_RT option (see [Section 23.23](#)) and any other options the client is interested in receiving. The client MAY additionally include instances of those options that are identified in the Option Request option, with data values as hints to the server about parameter values the client would like to have returned.

The client includes a Reconfigure Accept option (see [Section 23.20](#)) if the client is willing to accept Reconfigure messages from the server.



### **18.1.2. Transmission of Solicit Messages**

The first Solicit message from the client on the interface MUST be delayed by a random amount of time between 0 and SOL\_MAX\_DELAY. In the case of a Solicit message transmitted when DHCP is initiated by IPv6 Neighbor Discovery, the delay gives the amount of time to wait after IPv6 Neighbor Discovery causes the client to invoke the stateful address autoconfiguration protocol (see [section 5.5.3 of \[RFC4862\]](#)). This random delay desynchronizes clients which start at the same time (for example, after a power outage).

The client transmits the message according to [Section 15](#), using the following parameters:

IRT	SOL_TIMEOUT
MRT	SOL_MAX_RT
MRC	0
MRD	0

If the client has included a Rapid Commit option in its Solicit message, the client terminates the waiting process as soon as a Reply message with a Rapid Commit option is received.

If the client is waiting for an Advertise message, the mechanism in [Section 15](#) is modified as follows for use in the transmission of Solicit messages. The message exchange is not terminated by the receipt of an Advertise before the first RT has elapsed. Rather, the client collects Advertise messages until the first RT has elapsed. Also, the first RT MUST be selected to be strictly greater than IRT by choosing RAND to be strictly greater than 0.

A client MUST collect Advertise messages for the first RT seconds, unless it receives an Advertise message with a preference value of 255. The preference value is carried in the Preference option ([Section 23.8](#)). Any Advertise that does not include a Preference option is considered to have a preference value of 0. If the client receives an Advertise message that includes a Preference option with a preference value of 255, the client immediately begins a client-initiated message exchange (as described in [Section 19](#)) by sending a Request message to the server from which the Advertise message was received. If the client receives an Advertise message that does not include a Preference option with a preference value of 255, the client continues to wait until the first RT elapses. If the first RT elapses and the client has received an Advertise message, the client



SHOULD continue with a client-initiated message exchange by sending a Request message.

If the client does not receive any Advertise messages before the first RT has elapsed, it begins the retransmission mechanism described in [Section 15](#). The client terminates the retransmission process as soon as it receives any Advertise message, and the client acts on the received Advertise message without waiting for any additional Advertise messages.

A DHCP client SHOULD choose MRC and MRD to be 0. If the DHCP client is configured with either MRC or MRD set to a value other than 0, it MUST stop trying to configure the interface if the message exchange fails. After the DHCP client stops trying to configure the interface, it SHOULD restart the reconfiguration process after some external event, such as user input, system restart, or when the client is attached to a new link.

#### **[18.1.3](#). Receipt of Advertise Messages**

The client MUST process SOL\_MAX\_RT and INF\_MAX\_RT options in an Advertise message, even if the message contains a Status Code option indicating a failure, and the Advertise message will be discarded by the client.

The client MUST ignore any Advertise message that contains no addresses (IAADDR options encapsulated in IA\_NA or IA\_TA options) and no delegated prefixes (IAPREFIX options encapsulated in IA\_PD options) with the exception that the client:

- MUST process an included SOL\_MAX\_RT option and
- MUST process an included INF\_MAX\_RT option.

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

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

Upon receipt of one or more valid Advertise messages, the client selects one or more Advertise messages based upon the following criteria.

- Those Advertise messages with the highest server preference value are preferred over all other Advertise messages.



- Within a group of Advertise messages with the same server preference value, a client MAY select those servers whose Advertise messages advertise information of interest to the client.
- The client MAY choose a less-preferred server if that server has a better set of advertised parameters, such as the available set of IAs, as well as the set of other configuration options advertised.

Once a client has selected Advertise message(s), the client will typically store information about each server, such as server preference value, addresses advertised, when the advertisement was received, and so on.

In practice, this means that the client will maintain independent per-IA state machines per each selected server.

If the client needs to select an alternate server in the case that a chosen server does not respond, the client chooses the next server according to the criteria given above.

#### **18.1.4. Receipt of Reply Message**

If the client includes a Rapid Commit option in the Solicit message, it will expect a Reply message that includes a Rapid Commit option in response. The client discards any Reply messages it receives that do not include a Rapid Commit option. If the client receives a valid Reply message that includes a Rapid Commit option, it processes the message as described in [Section 19.1.8](#). If it does not receive such a Reply message and does receive a valid Advertise message, the client processes the Advertise message as described in [Section 18.1.3](#).

If the client subsequently receives a valid Reply message that includes a Rapid Commit option, it either:

- processes the Reply message as described in [Section 19.1.8](#), and discards any Reply messages received in response to the Request message, or
- processes any Reply messages received in response to the Request message and discards the Reply message that includes the Rapid Commit option.





## **[18.2.](#) Server Behavior**

A server sends an Advertise message in response to valid Solicit messages it receives to announce the availability of the server to the client.

### **[18.2.1.](#) Receipt of Solicit Messages**

The server determines the information about the client and its location as described in [Section 12](#) and checks its administrative policy about responding to the client. If the server is not permitted to respond to the client, the server discards the Solicit message. For example, if the administrative policy for the server is that it may only respond to a client that is willing to accept a Reconfigure message, if the client does not include a Reconfigure Accept option (see [Section 23.20](#)) in the Solicit message, the servers discard the Solicit message.

If the client has included a Rapid Commit option in the Solicit message and the server has been configured to respond with committed lease assignments and other resources, the server responds to the Solicit with a Reply message as described in [Section 18.2.3](#). Otherwise, the server ignores the Rapid Commit option and processes the remainder of the message as if no Rapid Commit option were present.

### **[18.2.2.](#) Creation and Transmission of Advertise Messages**

The server sets the "msg-type" field to ADVERTISE and copies the contents of the transaction-id field from the Solicit message received from the client to the Advertise message. The server includes its server identifier in a Server Identifier option and copies the Client Identifier from the Solicit message into the Advertise message.

The server MAY add a Preference option to carry the preference value for the Advertise message. The server implementation SHOULD allow the setting of a server preference value by the administrator. The server preference value MUST default to zero unless otherwise configured by the server administrator.

The server includes a Reconfigure Accept option if the server wants to require that the client accept Reconfigure messages.

The server includes options the server will return to the client in a subsequent Reply message. The information in these options may be used by the client in the selection of a server if the client receives more than one Advertise message. If the client has included



an Option Request option in the Solicit message, the server includes options in the Advertise message containing configuration parameters for all of the options identified in the Option Request option that the server has been configured to return to the client. The server MAY return additional options to the client if it has been configured to do so. The server must be aware of the recommendations on packet sizes and the use of fragmentation in [section 5 of \[RFC2460\]](#).

If the Solicit message from the client included one or more IA options, the server MUST include IA options in the Advertise message containing any addresses and/or delegated prefixes that would be assigned to IAs contained in the Solicit message from the client. If the client has included addresses in the IA in the Solicit message, the server MAY use those addresses as hints about the addresses that the client would like to receive. If the client has included IA Prefix option in the IA\_PD, the server MAY use the prefix contained in the IPv6 prefix field and/or the prefix length contained in the "prefix-length" field as a hints about the prefixes the client would like to receive. If the server is not going to assign an address or delegated prefix received as a hint in the Solicit message, the server MUST NOT include this address or delegated prefix in the Advertise message

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

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

If the Solicit message was received directly by the server, the server unicasts the Advertise message directly to the client using the address in the source address field from the IP datagram in which the Solicit message was received. The Advertise message MUST be unicast on the link from which the Solicit message was received.

If the Solicit message was received in a Relay-forward message, the server constructs a Relay-reply message with the Advertise message in the payload of a "relay-message" option. If the Relay-forward messages included an Interface-id option, the server copies that option to the Relay-reply message. The server unicasts the Relay-reply message directly to the relay agent using the address in the source address field from the IP datagram in which the Relay-forward message was received.



### **18.2.3. Creation and Transmission of Reply Messages**

The server MUST commit the assignment of any addresses or other configuration information message before sending a Reply message to a client in response to a Solicit message.

#### **DISCUSSION:**

When using the Solicit-Reply message exchange, the server commits the assignment of any leases before sending the Reply message. The client can assume it has been assigned the leases in the Reply message and does not need to send a Request message for those addresses.

Typically, servers that are configured to use the Solicit-Reply message exchange will be deployed so that only one server will respond to a Solicit message. If more than one server responds, the client will only use the leases from one of the servers, while the leases from the other servers will be committed to the client but not used by the client.

The server includes a Rapid Commit option in the Reply message to indicate that the Reply is in response to a Solicit message.

The server includes a Reconfigure Accept option if the server wants to require that the client accept Reconfigure messages.

The server produces the Reply message as though it had received a Request message, as described in [Section 19.2.1](#). The server transmits the Reply message as described in [Section 19.2.8](#).

## **19. DHCP Client-Initiated Configuration Exchange**

A client initiates a message exchange with a server or servers to acquire or update configuration information of interest. The client may initiate the configuration exchange as part of the operating system configuration process, when requested to do so by the application layer, when required by Stateless Address Autoconfiguration or as required to extend the lifetime of address(es) and/or delegated prefix(es), using Renew and Rebind messages.

According to a terminology for the prefix delegation, a client requesting a delegation of a prefix is referred to as a requesting router and a server delegating the prefix is referred to as a delegating router. The requesting router and the delegating router use the IA\_PD Prefix option to exchange information about prefix(es) in much the same way as IA Address options are used for assigned



addresses. Typically, a single DHCP session is used to exchange information about addresses and prefixes, i.e. IA\_NA and IA\_PD options are carried in the same message.

### **19.1. Client Behavior**

A client uses Request, Renew, Rebind, Release and Decline messages during the normal life cycle of addresses and delegated prefixes. When a client detects it may have moved to a new link, it uses Confirm if it only has addresses and Rebind if it has delegated prefixes (and addresses). It uses Information-request messages when it needs configuration information but no addresses and no prefixes.

If the client has a source address of sufficient scope that can be used by the server as a return address, and the client has received a Server Unicast option ([Section 23.12](#)) from the server, the client SHOULD unicast any Request, Renew, Release and Decline messages to the server.

#### **DISCUSSION:**

Use of unicast may avoid delays due to the relaying of messages by relay agents, as well as avoid overhead and duplicate responses by servers due to the delivery of client messages to multiple servers. Requiring the client to relay all DHCP messages through a relay agent enables the inclusion of relay agent options in all messages sent by the client. The server should enable the use of unicast only when relay agent options will not be used.

#### **19.1.1. Creation and Transmission of Request Messages**

The client uses a Request message to populate IAs with leases and obtain other configuration information. The client includes one or more IA options in the Request message. The server then returns leases and other information about the IAs to the client in IA options in a Reply message.

The client generates a transaction ID and inserts this value in the "transaction-id" field.

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

The client MUST include a Client Identifier option to identify itself to the server. The client adds any other appropriate options, including one or more IA options (if the client is requesting that the server assign it some network addresses or delegated prefixes).





The client MUST include an Option Request option (see [Section 23.7](#)) to indicate the options the client is interested in receiving. The client MAY include options with data values as hints to the server about parameter values the client would like to have returned.

The client includes a Reconfigure Accept option (see [Section 23.20](#)) indicating whether or not the client is willing to accept Reconfigure messages from the server.

The client transmits the message according to [Section 15](#), using the following parameters:

IRT	REQ_TIMEOUT
MRT	REQ_MAX_RT
MRC	REQ_MAX_RC
MRD	0

If the message exchange fails, the client takes an action based on the client's local policy. Examples of actions the client might take include:

- Select another server from a list of servers known to the client; for example, servers that responded with an Advertise message.
- Initiate the server discovery process described in [Section 18](#).
- Terminate the configuration process and report failure.

#### **[19.1.2](#). Creation and Transmission of Confirm Messages**

Whenever a client may have moved to a new link, the prefixes/addresses assigned to the interfaces on that link may no longer be appropriate for the link to which the client is attached. Examples of times when a client may have moved to a new link include:

- o The client reboots (and has no stable storage or persisted DHCP state).
- o The client is physically connected to a wired connection.
- o The client returns from sleep mode.
- o The client using a wireless technology changes access points.



In any situation when a client may have moved to a new link and the client does not have any delegated prefixes obtained from the DHCP server from which it has obtained the addresses, the client SHOULD initiate a Confirm/Reply message exchange. The client includes any IAs assigned to the interface that may have moved to a new link, along with the addresses associated with those IAs, in its Confirm message. Any responding servers will indicate whether those addresses are appropriate for the link to which the client is attached with the status in the Reply message it returns to the client.

If the client has any valid delegated prefixes obtained from the DHCP server from which it has obtained the addresses, the client initiates Rebind/Reply exchange as described in [Section 19.1.9](#) instead of sending the Confirm message.

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

The client MUST include a Client Identifier option to identify itself to the server. The client includes IA options for all of the IAs assigned to the interface for which the Confirm message is being sent. The IA options include all of the addresses the client currently has associated with those IAs. The client SHOULD set the T1 and T2 fields in any IA\_NA options and the preferred-lifetime and valid-lifetime fields in the IA Address options to 0, as the server will ignore these fields.

The first Confirm message from the client on the interface MUST be delayed by a random amount of time between 0 and CNF\_MAX\_DELAY. The client transmits the message according to [Section 15](#), using the following parameters:

IRT	CNF_TIMEOUT
MRT	CNF_MAX_RT
MRC	0
MRD	CNF_MAX_RD

If the client receives no responses before the message transmission process terminates, as described in [Section 15](#), the client SHOULD continue to use any IP addresses, using the last known lifetimes for those addresses, and SHOULD continue to use any other previously obtained configuration parameters.



### **19.1.3. Creation and Transmission of Renew Messages**

To extend the valid and preferred lifetimes for the leases assigned to the IAs, the client sends a Renew message to the server from which the leases were obtained, which includes IA options for the IAs whose lease lifetimes are to be extended. The client includes IA Address options within IA\_NA and IA\_TA options for the addresses assigned to the IAs. The client includes IA Prefix options within IA\_PD options for the delegated prefixes assigned to the IAs. The server determines new lifetimes for the leases according to the administrative configuration of the server. The server may also add leases to the IAs. The server can remove leases from the IAs by returning IA Address options (for IA\_NA and IA\_TA) and IA Prefix options (for IA\_PD) with preferred and valid lifetimes set to 0.

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

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

If T1 or T2 had been set to 0 by the server (for an IA\_NA or IA\_PD) or there are no T1 or T2 times (for an IA\_TA) in a previous Reply, the client may send a Renew or Rebind message, respectively, at the client's discretion. The client **MUST** follow the rules defined in [Section 14.2](#).

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

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

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

For IAs to which leases have been assigned, the client includes a corresponding IA option containing an IA Address option for each address assigned to the IA and IA Prefix option for each prefix assigned to the IA. The client **MUST NOT** include addresses and



prefixes in any IA option that the client did not obtain from the server or that are no longer valid (that have a valid lifetime of 0).

The client MAY include an IA option for each binding it desires but has been unable to obtain. In this case, if the client includes the IA\_PD option to request prefix delegation, the client MAY include the IA Prefix option encapsulated within the IA\_PD option, with the IPv6 prefix field set to 0 and the "prefix-length" field set to the desired length of the prefix to be delegated. The server MAY use this value as a hint for the prefix length. The client SHOULD NOT include IA Prefix option with the IPv6 prefix field set to 0 unless it is supplying a hint for the prefix length.

The client MUST include an Option Request option (see [Section 23.7](#)) to indicate the options the client is interested in receiving. The client MAY include options with data values as hints to the server about parameter values the client would like to have returned.

The client transmits the message according to [Section 15](#), using the following parameters:

IRT	REN_TIMEOUT
MRT	REN_MAX_RT
MRC	0
MRD	Remaining time until T2

The message exchange is terminated when time T2 is reached (see [Section 19.1.4](#)), at which time the client begins a Rebind message exchange.

#### **[19.1.4](#). Creation and Transmission of Rebind Messages**

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

The client constructs the Rebind message as described in [section 18.1.3](#) with the following differences:

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





The client transmits the message according to [Section 15](#), using the following parameters:

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

If all leases for an IA have expired, the client may choose to include this IA in subsequent Rebind messages to indicate that the client is interested in assignment of the leases to this IA.

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

#### **[19.1.5](#). Creation and Transmission of Information-request Messages**

The client uses an Information-request message to obtain configuration information without having addresses and/or delegated prefixes assigned to it.

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

The client SHOULD include a Client Identifier option to identify itself to the server. If the client does not include a Client Identifier option, the server will not be able to return any client-specific options to the client, or the server may choose not to respond to the message at all. The client MUST include a Client Identifier option if the Information-request message will be authenticated.

The client MUST include an Option Request option (see [Section 23.7](#)) to request the INF\_MAX\_RT option (see [Section 23.24](#)) and any other options the client is interested in receiving. The client MAY include options with data values as hints to the server about parameter values the client would like to have returned.

The first Information-request message from the client on the interface MUST be delayed by a random amount of time between 0 and



INF\_MAX\_DELAY. The client transmits the message according to [Section 15](#), using the following parameters:

IRT	INF_TIMEOUT
MRT	INF_MAX_RT
MRC	0
MRD	0

#### **[19.1.6](#). Creation and Transmission of Release Messages**

To release one or more leases, a client sends a Release message to the server.

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

The client places the identifier of the server that allocated the lease(s) in a Server Identifier option.

The client MUST include a Client Identifier option to identify itself to the server. The client includes options containing the IAs for the leases it is releasing in the "options" field. The leases to be released MUST be included in the IAs. Any leases for the IAs the client wishes to continue to use MUST NOT be added to the IAs.

The client MUST NOT use any of the addresses it is releasing as the source address in the Release message or in any subsequently transmitted message.

Because Release messages may be lost, the client should retransmit the Release if no Reply is received. However, there are scenarios where the client may not wish to wait for the normal retransmission timeout before giving up (e.g., on power down). Implementations SHOULD retransmit one or more times, but MAY choose to terminate the retransmission procedure early.

The client transmits the message according to [Section 15](#), using the following parameters:

IRT	REL_TIMEOUT
MRT	0
MRC	REL_MAX_RC



MRD      0

The client MUST stop using all of the leases being released as soon as the client begins the Release message exchange process. If leases are released but the Reply from a DHCP server is lost, the client will retransmit the Release message, and the server may respond with a Reply indicating a status of NoBinding. Therefore, the client does not treat a Reply message with a status of NoBinding in a Release message exchange as if it indicates an error.

Note that if the client fails to release the lease, each lease assigned to the IA will be reclaimed by the server when the valid lifetime of that lease expires.

#### **19.1.7.    Creation and Transmission of Decline Messages**

If a client detects that one or more addresses assigned to it by a server are already in use by another node, the client sends a Decline message to the server to inform it that the address is suspect.

The Decline message is not used in prefix delegation and thus the client MUST NOT include IA\_PD options in the Decline message.

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

The client places the identifier of the server that allocated the address(es) in a Server Identifier option.

The client MUST include a Client Identifier option to identify itself to the server. The client includes options containing the IAs for the addresses it is declining in the "options" field. The addresses to be declined MUST be included in the IAs. Any addresses for the IAs the client wishes to continue to use should not be included to the IAs.

The client MUST NOT use any of the addresses it is declining as the source address in the Decline message or in any subsequently transmitted message.

The client transmits the message according to [Section 15](#), using the following parameters:

IRT      DEC\_TIMEOUT

MRT      0



MRC      DEC\_MAX\_RC

MRD      0

If addresses are declined but the Reply from a DHCP server is lost, the client will retransmit the Decline message, and the server may respond with a Reply indicating a status of NoBinding. Therefore, the client does not treat a Reply message with a status of NoBinding in a Decline message exchange as if it indicates an error.

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

#### **19.1.8.    Receipt of Reply Messages**

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

If the client receives a Reply message with a status code containing UnspecFail, the server is indicating that it was unable to process the message due to an unspecified failure condition. If the client retransmits the original message to the same server to retry the desired operation, the client MUST limit the rate at which it retransmits the message and limit the duration of the time during which it retransmits the message (see [Section 14.1](#)).

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

When the client receives a NotOnLink status from the server in response to a Confirm message, the client performs DHCP server solicitation, as described in [Section 18](#), and client-initiated configuration, as described in [Section 19](#). If the client receives any Reply messages that do not indicate a NotOnLink status, the client can use the addresses in the IA and ignore any messages that indicate a NotOnLink status.





When the client receives a NotOnLink status from the server in response to a Solicit (with a Rapid Commit option) or a Request, the client can either re-issue the message without specifying any addresses or restart the DHCP server discovery process (see [Section 18](#)).

The client SHOULD perform duplicate address detection [[RFC4862](#)] on each of the received addresses in any IAs, on which it has not performed duplicate address detection during processing of any of the previous Reply messages from the server. The client performs the duplicate address detection before using the received addresses for the traffic. If any of the addresses are found to be in use on the link, the client sends a Decline message to the server for those addresses as described in [Section 19.1.7](#).

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

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

Management of the specific configuration information is detailed in the definition of each option in [Section 23](#).

The client examines the status code in each IA individually. If the client receives a NoAddrsAvail and/or NoPrefixAvail status code, the client has received no usable addresses and/or delegated prefixes in the IA.

If the client can operate with the addresses and prefixes obtained from the server, the client uses the addresses, delegated prefixes, and other information from any IAs that do not contain a Status Code option with the NoAddrsAvail or NoPrefixAvail status code. The client MAY include the IAs for which it received the NoAddrsAvail or



NoPrefixAvail status code, with no addresses or prefixes, in subsequent Renew and Rebind messages sent to the server, to retry obtaining the addresses or prefixes for these IAs.

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

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

Whenever a client restarts the DHCP server discovery process or selects an alternate server, as described in [Section 18.1.3](#), the client SHOULD stop using all the addresses and delegated prefixes for which it has the bindings and try to obtain all required leases from the new server. This facilitates the client using a single state machine for all bindings.

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

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

When the requesting router receives a valid Reply message, for each IA\_PD the delegating router assigns a subnet from each of the delegated prefixes to each of the links to which the associated interfaces are attached.

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



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

#### **[19.1.9.](#) Bindings Verification by Requesting Router**

In some circumstances the requesting router may need verification that the delegating router still has a valid binding for the requesting router. Examples of times when a requesting router may ask for such verification include:

- o The requesting router reboots.
- o The requesting router's upstream link flaps.
- o The requesting router is physically disconnected from a wired connection.

If such verification is needed the requesting router MUST initiate a Rebind/Reply message exchange as described in [Section 19.1.4](#), with the exception that the retransmission parameters should be set as for the Confirm message, described in [Section 19.1.2](#). The requesting router includes any IA\_PDs, along with prefixes associated with those IA\_PDs in its Rebind message.

#### **[19.2.](#) Server Behavior**

For this discussion, the Server is assumed to have been configured in an implementation specific manner with configuration of interest to clients.

In most instances, the server will send a Reply in response to a client message. This Reply message MUST always contain the Server Identifier option containing the server's DUID and the Client Identifier option from the client message if one was present.

In most Reply messages, the server includes options containing configuration information for the client. The server must be aware of the recommendations on packet sizes and the use of fragmentation in [section 5 of \[RFC2460\]](#). If the client included an Option Request option in its message, the server includes options in the Reply message containing configuration parameters for all of the options identified in the Option Request option that the server has been configured to return to the client. The server MAY return additional options to the client if it has been configured to do so.



### **19.2.1. Receipt of Request Messages**

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

When the server receives a valid Request message, the server creates the bindings for that client according to the server's policy and configuration information and records the IAs and other information requested by the client.

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

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

The server examines all IAs in the message from the client.

For each IA\_NA and IA\_TA the server checks if the prefixes on included IP addresses are appropriate for the link to which the client is connected. If any of the prefixes on the included IP addresses is not appropriate for the link to which the client is connected, the server MUST return the IA to the client with a Status Code option with the value NotOnLink. If the server does not send the NotOnLink status code but it cannot assign any IP addresses to an IA, the server MUST return the IA in the Reply message with no addresses in the IA and a Status Code option containing status code NoAddrsAvail.

For any IA\_PD to which the server cannot assign any delegated prefixes, the server MUST return the IA\_PD option in the Reply message with the Status Code option containing status code NoPrefixAvail.

The server MAY assign different addresses and/or delegated prefixes to an IA than included in the IA within the Request message sent by the client.

For all IAs to which the server can assign addresses or delegated prefixes, the server includes the IAs with addresses (for IA\_NA and IA\_TA), prefixes (for IA\_PD) and other configuration parameters, and





records the IA as a new client binding. The server MUST NOT include any addresses or delegated prefixes in the IA which the server does not assign to the client.

The server includes a Reconfigure Accept option if the server wants to require that the client accept Reconfigure messages.

The server includes other options containing configuration information to be returned to the client as described in [Section 19.2](#).

If the server finds that the client has included an IA in the Request message for which the server already has a binding that associates the IA with the client, the client has resent a Request message for which it did not receive a Reply message. The server either resends a previously cached Reply message or sends a new Reply message.

#### **[19.2.2](#). Receipt of Confirm Messages**

When the server receives a Confirm message, the server determines whether the addresses in the Confirm message are appropriate for the link to which the client is attached. If all of the addresses in the Confirm message pass this test, the server returns a status of Success. If any of the addresses do not pass this test, the server returns a status of NotOnLink. If the server is unable to perform this test (for example, the server does not have information about prefixes on the link to which the client is connected), or there were no addresses in any of the IAs sent by the client, the server MUST NOT send a Reply to the client.

The server ignores the T1 and T2 fields in the IA options and the preferred-lifetime and valid-lifetime fields in the IA Address options.

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

The server MUST include a Server Identifier option containing the server's DUID and the Client Identifier option from the Confirm message in the Reply message. The server includes a Status Code option indicating the status of the Confirm message.

#### **[19.2.3](#). Receipt of Renew Messages**

When the server receives a Renew message via unicast from a client to which the server has not sent a unicast option (or is not currently configured to send a unicast option to the client), the server



discards the Renew message and responds with a Reply message containing a Status Code option with the value UseMulticast, a Server Identifier option containing the server's DUID, the Client Identifier option from the client message, and no other options.

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

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

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

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

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

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

- If the server is configured to create new bindings as a result of processing Renew messages, the server SHOULD create a binding and return the IA with assigned addresses or delegated prefixes with lifetimes and, if applicable, T1/T2 times and other information requested by the client. If the client included the IA Prefix option within the IA\_PD option with the non-zero value in the "prefix-length" field, the server MAY use this value as a hint for the length of the prefixes to be assigned (see [\[I-D.cui-dhc-dhcpv6-prefix-length-hint-issue\]](#) for further details on prefix length hints).
- If the server is configured to create new bindings as a result of processing Renew messages, but the server will not assign any leases to an IA, the server returns the IA option containing a Status Code option with the NoAddrsAvail or NoPrefixAvail status code and a status message for a user.



- If the server does not support creation of new bindings for the client sending a Renew message, or if this behavior is disabled according to the server's policy or configuration information, the server returns the IA option containing a Status Code option with the NoBinding status code and a status message for a user.

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

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

The server includes other options containing configuration information to be returned to the client as described in [Section 19.2](#).

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

#### **[19.2.4](#). Receipt of Rebind Messages**

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

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

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



If the server cannot find a client entry for the IA, the server checks if the IA contains addresses (for IA\_NA and IA\_TA) or delegated prefixes (for IA\_PD). The server checks if the addresses and delegated prefixes are appropriate for the link to which the client's interface is attached according to the server's explicit configuration information. For any address which is not appropriate for the link to which the client's interface is attached, the server MAY include the IA Address option with the lifetimes of 0. For any delegated prefix which is not appropriate for the link to which the client's interface is attached, the server MAY include the IA Prefix option with the lifetimes of 0. The Reply with lifetimes of 0 constitutes an explicit notification to the client that the specific addresses and delegated prefixes are no longer valid and MUST NOT be used by the client. In this situation, if the server does not send a Reply message, it silently discards the Rebind message.

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

- If the server is configured to create new bindings as a result of processing Rebind messages (also see the note about the Rapid Commit option below), the server SHOULD create a binding and return the IA with allocated leases with lifetimes and, if applicable, T1/T2 times and other information requested by the client. The server MUST NOT return any addresses or delegated prefixes in the IA which the server does not assign to the client.
- If the server is configured to create new bindings as a result of processing Rebind messages, but the server will not assign any leases to an IA, the server returns the IA option containing a Status Code option with the NoAddrsAvail or NoPrefixAvail status code and a status message for a user.
- If the server does not support creation of new bindings for the client sending a Rebind message, or if this behavior is disabled according to the server's policy or configuration information, the server returns the IA option containing a Status Code option with the NoBinding status code and a status message for a user.

When the server creates new bindings for the IA, it is possible that other servers also create bindings as a result of receiving the same Rebind message. This is the same issue as in the Discussion under "Rapid Commit Option"; see [Section 23.14](#). Therefore, the server SHOULD only create new bindings during processing of a Rebind message if the server is configured to respond with a Reply message to a Solicit message containing the Rapid Commit option.





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

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

The server includes other options containing configuration information to be returned to the client as described in [Section 19.2](#).

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

#### **[19.2.5](#). Receipt of Information-request Messages**

When the server receives an Information-request message, the client is requesting configuration information that does not include the assignment of any leases. The server determines all configuration parameters appropriate to the client, based on the server configuration policies known to the server.

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

The server MUST include a Server Identifier option containing the server's DUID in the Reply message. If the client included a Client Identification option in the Information-request message, the server copies that option to the Reply message.

The server includes options containing configuration information to be returned to the client as described in [Section 19.2](#).

If the Information-request message received from the client did not include a Client Identifier option, the server SHOULD respond with a Reply message containing any configuration parameters that are not determined by the client's identity. If the server chooses not to respond, the client may continue to retransmit the Information-request message indefinitely.



#### **19.2.6. Receipt of Release Messages**

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

Upon the receipt of a valid Release message, the server examines the IAs and the leases in the IAs for validity. If the IAs in the message are in a binding for the client, and the leases in the IAs have been assigned by the server to those IAs, the server deletes the leases from the IAs and makes the leases available for assignment to other clients. The server ignores leases not assigned to the IA, although it may choose to log an error.

After all the leases have been processed, the server generates a Reply message and includes a Status Code option with value Success, a Server Identifier option with the server's DUID, and a Client Identifier option with the client's DUID. For each IA in the Release message for which the server has no binding information, the server adds an IA option using the IAID from the Release message, and includes a Status Code option with the value NoBinding in the IA option. No other options are included in the IA option.

A server may choose to retain a record of assigned leases and IAs after the lifetimes on the leases have expired to allow the server to reassign the previously assigned leases to a client.

#### **19.2.7. Receipt of Decline Messages**

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

Upon the receipt of a valid Decline message, the server examines the IAs and the addresses in the IAs for validity. If the IAs in the message are in a binding for the client, and the addresses in the IAs have been assigned by the server to those IAs, the server deletes the addresses from the IAs. The server ignores addresses not assigned to the IA (though it may choose to log an error if it finds such an address).



The client has found any addresses in the Decline messages to be already in use on its link. Therefore, the server SHOULD mark the addresses declined by the client so that those addresses are not assigned to other clients, and MAY choose to make a notification that addresses were declined. Local policy on the server determines when the addresses identified in a Decline message may be made available for assignment.

After all the addresses have been processed, the server generates a Reply message and includes a Status Code option with the value Success, a Server Identifier option with the server's DUID, and a Client Identifier option with the client's DUID. For each IA in the Decline message for which the server has no binding information, the server adds an IA option using the IAID from the Decline message and includes a Status Code option with the value NoBinding in the IA option. No other options are included in the IA option.

#### **[19.2.8](#). Transmission of Reply Messages**

If the original message was received directly by the server, the server unicasts the Reply message directly to the client using the address in the source address field from the IP datagram in which the original message was received. The Reply message MUST be unicast through the interface on which the original message was received.

If the original message was received in a Relay-forward message, the server constructs a Relay-reply message with the Reply message in the payload of a Relay Message option (see [Section 23.10](#)). If the Relay-forward messages included an Interface-id option, the server copies that option to the Relay-reply message. The server unicasts the Relay-reply message directly to the relay agent using the address in the source address field from the IP datagram in which the Relay-forward message was received.

### **[20](#). DHCP Server-Initiated Configuration Exchange**

A server initiates a configuration exchange to cause DHCP clients to obtain new addresses and other configuration information. For example, an administrator may use a server-initiated configuration exchange when links in the DHCP domain are to be renumbered. Other examples include changes in the location of directory servers, addition of new services such as printing, and availability of new software.



## **[20.1.](#) Server Behavior**

A server sends a Reconfigure message to cause a client to initiate immediately a Renew/Reply or Information-request/Reply message exchange with the server.

### **[20.1.1.](#) Creation and Transmission of Reconfigure Messages**

The server sets the "msg-type" field to RECONFIGURE. The server sets the transaction-id field to 0. The server includes a Server Identifier option containing its DUID and a Client Identifier option containing the client's DUID in the Reconfigure message.

The server MAY include an Option Request option to inform the client of what information has been changed or new information that has been added. In particular, the server specifies the IA option in the Option Request option if the server wants the client to obtain new address information. If the server identifies the IA option in the Option Request option, the server MUST include an IA option to identify each IA that is to be reconfigured on the client. The IA options included by the server MUST NOT contain any options.

Because of the risk of denial of service attacks against DHCP clients, the use of a security mechanism is mandated in Reconfigure messages. The server MUST use DHCP authentication in the Reconfigure message.

The server MUST include a Reconfigure Message option (defined in [Section 23.19](#)) to select whether the client responds with a Renew message, a Rebind message, or an Information-request message.

The server MUST NOT include any other options in the Reconfigure except as specifically allowed in the definition of individual options.

A server sends each Reconfigure message to a single DHCP client, using an IPv6 unicast address of sufficient scope belonging to the DHCP client. If the server does not have an address to which it can send the Reconfigure message directly to the client, the server uses a Relay-reply message (as described in [Section 21.3](#)) to send the Reconfigure message to a relay agent that will relay the message to the client. The server may obtain the address of the client (and the appropriate relay agent, if required) through the information the server has about clients that have been in contact with the server, or through some external agent.

To reconfigure more than one client, the server unicasts a separate message to each client. The server may initiate the reconfiguration





of multiple clients concurrently; for example, a server may send a Reconfigure message to additional clients while previous reconfiguration message exchanges are still in progress.

The Reconfigure message causes the client to initiate a Renew/Reply, a Rebind/Reply, or Information-request/Reply message exchange with the server. The server interprets the receipt of a Renew, a Rebind, or Information-request message (whichever was specified in the original Reconfigure message) from the client as satisfying the Reconfigure message request.

#### **[20.1.2.](#) Time Out and Retransmission of Reconfigure Messages**

If the server does not receive a Renew, Rebind, or Information-request message from the client in REC\_TIMEOUT milliseconds, the server retransmits the Reconfigure message, doubles the REC\_TIMEOUT value and waits again. The server continues this process until REC\_MAX\_RC unsuccessful attempts have been made, at which point the server SHOULD abort the reconfigure process for that client.

Default and initial values for REC\_TIMEOUT and REC\_MAX\_RC are documented in [Section 6.5](#).

#### **[20.2.](#) Receipt of Renew or Rebind Messages**

In response to a Renew message, the server generates and sends a Reply message to the client as described in [Section 19.2.3](#) and [Section 19.2.8](#), including options for configuration parameters.

In response to a Rebind message, the server generates and sends a Reply message to the client as described in [Section 19.2.4](#) and [Section 19.2.8](#), including options for configuration parameters.

The server MAY include options containing the IAs and new values for other configuration parameters in the Reply message, even if those IAs and parameters were not requested in the Renew or Rebind message from the client.

#### **[20.3.](#) Receipt of Information-request Messages**

The server generates and sends a Reply message to the client as described in [Section 19.2.5](#) and [Section 19.2.8](#), including options for configuration parameters.

The server MAY include options containing new values for other configuration parameters in the Reply message, even if those parameters were not requested in the Information-request message from the client.



## **20.4. Client Behavior**

A client receives Reconfigure messages sent to the UDP port 546 on interfaces for which it has acquired configuration information through DHCP. These messages may be sent at any time. Since the results of a reconfiguration event may affect application layer programs, the client **SHOULD** log these events, and **MAY** notify these programs of the change through an implementation-specific interface.

### **20.4.1. Receipt of Reconfigure Messages**

Upon receipt of a valid Reconfigure message, the client responds with either a Renew message, a Rebind message, or an Information-request message as indicated by the Reconfigure Message option (as defined in [Section 23.19](#)). The client ignores the transaction-id field in the received Reconfigure message. While the transaction is in progress, the client discards any Reconfigure messages it receives.

#### **DISCUSSION:**

The Reconfigure message acts as a trigger that signals the client to complete a successful message exchange. Once the client has received a Reconfigure, the client proceeds with the message exchange (retransmitting the Renew or Information-request message if necessary); the client ignores any additional Reconfigure messages until the exchange is complete. Subsequent Reconfigure messages cause the client to initiate a new exchange.

How does this mechanism work in the face of duplicated or retransmitted Reconfigure messages? Duplicate messages will be ignored because the client will begin the exchange after the receipt of the first Reconfigure. Retransmitted messages will either trigger the exchange (if the first Reconfigure was not received by the client) or will be ignored. The server can discontinue retransmission of Reconfigure messages to the client once the server receives the Renew or Information-request message from the client.

It might be possible for a duplicate or retransmitted Reconfigure to be sufficiently delayed (and delivered out of order) to arrive at the client after the exchange (initiated by the original Reconfigure) has been completed. In this case, the client would initiate a redundant exchange. The likelihood of delayed and out of order delivery is small enough to be ignored. The consequence of the redundant exchange is inefficiency rather than incorrect operation.



#### **[20.4.2.](#) Creation and Transmission of Renew or Rebind Messages**

When responding to a Reconfigure, the client creates and sends the Renew message in exactly the same manner as outlined in [Section 19.1.3](#), with the exception that the client copies the Option Request option and any IA options from the Reconfigure message into the Renew message. The client MUST include a Server Identifier option in the Renew message, identifying the server with which the client most recently communicated.

When responding to a Reconfigure, the client creates and sends the Rebind message in exactly the same manner as outlined in [Section 19.1.4](#), with the exception that the client copies the Option Request option and any IA options from the Reconfigure message into the Rebind message.

If a client is currently sending Rebind messages, as described in [Section 19.1.3](#), the client ignores any received Reconfigure messages.

#### **[20.4.3.](#) Creation and Transmission of Information-request Messages**

When responding to a Reconfigure, the client creates and sends the Information-request message in exactly the same manner as outlined in [Section 19.1.5](#), with the exception that the client includes a Server Identifier option with the identifier from the Reconfigure message to which the client is responding.

#### **[20.4.4.](#) Time Out and Retransmission of Renew, Rebind or Information-request Messages**

The client uses the same variables and retransmission algorithm as it does with Renew, Rebind, or Information-request messages generated as part of a client-initiated configuration exchange. See [Section 19.1.3](#), [Section 19.1.4](#), and [Section 19.1.5](#) for details. If the client does not receive a response from the server by the end of the retransmission process, the client ignores and discards the Reconfigure message.

#### **[20.4.5.](#) Receipt of Reply Messages**

Upon the receipt of a valid Reply message, the client processes the options and sets (or resets) configuration parameters appropriately. The client records and updates the lifetimes for any addresses specified in IAs in the Reply message.



## **[20.5.](#) Prefix Delegation Reconfiguration**

This section describes prefix delegation in Reconfigure message exchanges.

### **[20.5.1.](#) Delegating Router Behavior**

The delegating router initiates a configuration message exchange with a requesting router, as described in [Section 20](#), by sending a Reconfigure message (acting as a DHCP server) to the requesting router, as described in [Section 20.1](#). The delegating router specifies the IA\_PD option in the Option Request option to cause the requesting router to include an IA\_PD option to obtain new information about delegated prefix(es).

### **[20.5.2.](#) Requesting Router Behavior**

The requesting router responds to a Reconfigure message, acting as a DHCP client, received from a delegating router as described in [Section 20.4](#). The requesting router MUST include the IA\_PD Prefix option(s) (in an IA\_PD option) for prefix(es) that have been delegated to the requesting router by the delegating router from which the Reconfigure message was received.

## **[21.](#) Relay Agent Behavior**

The relay agent MAY be configured to use a list of destination addresses, which MAY include unicast addresses, the All\_DHCP\_Servers multicast address, or other addresses selected by the network administrator. If the relay agent has not been explicitly configured, it MUST use the All\_DHCP\_Servers multicast address as the default.

If the relay agent relays messages to the All\_DHCP\_Servers multicast address or other multicast addresses, it sets the Hop Limit field to 32.

If the relay agent receives a message other than Relay-forward and Relay-reply and the relay agent does not recognize its message type, it MUST forward them as described in [Section 21.1.1](#).

### **[21.1.](#) Relaying a Client Message or a Relay-forward Message**

A relay agent relays both messages from clients and Relay-forward messages from other relay agents. When a relay agent receives a valid message (for a definition of a valid message, see [Section 4.1 of \[RFC7283\]](#)) to be relayed, it constructs a new Relay-forward message. 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. The relay agent copies the received DHCP message (excluding any IP or UDP headers) into a Relay Message option in the new message. The relay agent adds to the Relay-forward message any other options it is configured to include.

[RFC6221] defines a Lightweight DHCPv6 Relay Agent (LDRA) that allows Relay Agent Information to be inserted by an access node that performs a link-layer bridging (i.e., non-routing) function.

#### **[21.1.1.](#) Relaying a Message from a Client**

If the relay agent received the message to be relayed from a client, the relay agent places a global, ULA [[RFC4193](#)] or site-scoped address with a prefix assigned to the link on which the client should be assigned an address in the link-address field. If not addresses of other scopes are available the relay agent may fill in the link-address field with a link-local address from the interface the original message was received on. That is not recommended as it requires additional information to be provided in the server configuration. See Section 3.2 of [[I-D.ietf-dhc-topo-conf](#)] for detailed discussion.

This address will be used by the server to determine the link from which the client should be assigned an address and other configuration information. The hop-count in the Relay-forward message is set to 0.

If the relay agent cannot use the address in the link-address field to identify the interface through which the response to the client will be relayed, the relay agent **MUST** include an Interface-id option (see [Section 23.18](#)) in the Relay-forward message. The server will include the Interface-id option in its Relay-reply message. The relay agent fills in the link-address field as described in the previous paragraph regardless of whether the relay agent includes an Interface-id option in the Relay-forward message.

#### **[21.1.2.](#) Relaying a Message from a Relay Agent**

If the message received by the relay agent is a Relay-forward message and the hop-count in the message is greater than or equal to HOP\_COUNT\_LIMIT, the relay agent discards the received message.

The relay agent copies the source address from the IP datagram in which the message was received from the relay agent into the peer-address field in the Relay-forward message and sets the hop-count field to the value of the hop-count field in the received message incremented by 1.



If the source address from the IP datagram header of the received message is a global or site-scoped address (and the device on which the relay agent is running belongs to only one site), the relay agent sets the link-address field to 0; otherwise the relay agent sets the link-address field to a global or site-scoped address assigned to the interface on which the message was received, or includes an Interface-ID option to identify the interface on which the message was received.

#### **21.1.3. Relay Agent Behavior with Prefix Delegation**

A relay agent forwards messages containing Prefix Delegation options in the same way as described earlier in this section.

If a delegating router communicates with a requesting router through a relay agent, the delegating router may need a protocol or other out-of-band communication to configure routing information for delegated prefixes on any router through which the requesting router may forward traffic.

#### **21.2. Relaying a Relay-reply Message**

The relay agent processes any options included in the Relay-reply message in addition to the Relay Message option, and then discards those options.

The relay agent extracts the message from the Relay Message option and relays it to the address contained in the peer-address field of the Relay-reply message. Relay agents **MUST NOT** modify the message.

If the Relay-reply message includes an Interface-id option, the relay agent relays the message from the server to the client on the link identified by the Interface-id option. Otherwise, if the link-address field is not set to zero, the relay agent relays the message on the link identified by the link-address field.

If the relay agent receives a Relay-reply message, it **MUST** process the message as defined above, regardless of the type of message encapsulated in the Relay Message option.

#### **21.3. Construction of Relay-reply Messages**

A server uses a Relay-reply message to return a response to a client if the original message from the client was relayed to the server in a Relay-forward message or to send a Reconfigure message to a client if the server does not have an address it can use to send the message directly to the client.



A response to the client **MUST** be relayed through the same relay agents as the original client message. The server causes this to happen by creating a Relay-reply message that includes a Relay Message option containing the message for the next relay agent in the return path to the client. The contained Relay-reply message contains another Relay Message option to be sent to the next relay agent, and so on. The server must record the contents of the peer-address fields in the received message so it can construct the appropriate Relay-reply message carrying the response from the server.

For example, if client C sent a message that was relayed by relay agent A to relay agent B and then to the server, the server would send the following Relay-Reply message to relay agent B:

```
msg-type:      RELAY-REPLY
hop-count:     1
link-address:   0
peer-address:   A
Relay Message option, containing:
  msg-type:     RELAY-REPLY
  hop-count:    0
  link-address: address from link to which C is attached
  peer-address: C
  Relay Message option: <response from server>
```

Figure 8: Relay-reply Example

When sending a Reconfigure message to a client through a relay agent, the server creates a Relay-reply message that includes a Relay Message option containing the Reconfigure message for the next relay agent in the return path to the client. The server sets the peer-address field in the Relay-reply message header to the address of the client, and sets the link-address field as required by the relay agent to relay the Reconfigure message to the client. The server obtains the addresses of the client and the relay agent through prior interaction with the client or through some external mechanism.

## **22. Authentication of DHCP Messages**

Some network administrators may wish to provide authentication of the source and contents of DHCP messages. For example, clients may be subject to denial of service attacks through the use of bogus DHCP servers, or may simply be misconfigured due to unintentionally instantiated DHCP servers. Network administrators may wish to constrain the allocation of addresses to authorized hosts to avoid denial of service attacks in "hostile" environments where the network



medium is not physically secured, such as wireless networks or college residence halls.

The DHCP authentication mechanism is based on the design of authentication for DHCPv4 [[RFC3118](#)].

### **22.1. Security of Messages Sent Between Servers and Relay Agents**

Relay agents and servers that exchange messages securely use the IPsec mechanisms for IPv6 [[RFC4301](#)]. If a client message is relayed through multiple relay agents, each of the relay agents must have established independent, pairwise trust relationships. That is, if messages from client C will be relayed by relay agent A to relay agent B and then to the server, relay agents A and B must be configured to use IPsec for the messages they exchange, and relay agent B and the server must be configured to use IPsec for the messages they exchange.

Relay agents and servers that support secure relay agent to server or relay agent to relay agent communication use IPsec under the following conditions:

Selectors	Relay agents are manually configured with the addresses of the relay agent or server to which DHCP messages are to be forwarded. Each relay agent and server that will be using IPsec for securing DHCP messages must also be configured with a list of the relay agents to which messages will be returned. The selectors for the relay agents and servers will be the pairs of addresses defining relay agents and servers that exchange DHCP messages on DHCPv6 UDP port 547.
Mode	Relay agents and servers use transport mode and ESP. The information in DHCP messages is not generally considered confidential, so encryption need not be used (i.e., NULL encryption can be used).
Key management	Because the relay agents and servers are used within an organization, public key schemes are not necessary. Because the relay agents and servers must be manually configured, manually configured key management may suffice, but does not provide defense against replayed messages. Accordingly, IKE with





preshared secrets SHOULD be supported. IKE with public keys MAY be supported.

Security policy	DHCP messages between relay agents and servers should only be accepted from DHCP peers as identified in the local configuration.
Authentication	Shared keys, indexed to the source IP address of the received DHCP message, are adequate in this application.
Availability	Appropriate IPsec implementations are likely to be available for servers and for relay agents in more featureful devices used in enterprise and core ISP networks. IPsec is less likely to be available for relay agents in low end devices primarily used in the home or small office markets.

## **22.2. Summary of DHCP Authentication**

Authentication of DHCP messages is accomplished through the use of the Authentication option (see [Section 23.11](#)). The authentication information carried in the Authentication option can be used to reliably identify the source of a DHCP message and to confirm that the contents of the DHCP message have not been tampered with.

The Authentication option provides a framework for multiple authentication protocols. Two such protocols are defined here. Other protocols defined in the future will be specified in separate documents.

Any DHCP message MUST NOT include more than one Authentication option.

The protocol field in the Authentication option identifies the specific protocol used to generate the authentication information carried in the option. The algorithm field identifies a specific algorithm within the authentication protocol; for example, the algorithm field specifies the hash algorithm used to generate the message authentication code (MAC) in the authentication option. The replay detection method (RDM) field specifies the type of replay detection used in the replay detection field.



### **[22.3.](#) Replay Detection**

The Replay Detection Method (RDM) field determines the type of replay detection used in the Replay Detection field.

If the RDM field contains 0x00, the replay detection field MUST be set to the value of a strictly monotonically increasing counter. Using a counter value, such as the current time of day (for example, an NTP-format timestamp [[RFC5905](#)]), can reduce the danger of replay attacks. This method MUST be supported by all protocols.

### **[22.4.](#) Reconfigure Key Authentication Protocol**

The Reconfigure key authentication protocol provides protection against misconfiguration of a client caused by a Reconfigure message sent by a malicious DHCP server. In this protocol, a DHCP server sends a Reconfigure Key to the client in the initial exchange of DHCP messages. The client records the Reconfigure Key for use in authenticating subsequent Reconfigure messages from that server. The server then includes an HMAC computed from the Reconfigure Key in subsequent Reconfigure messages.

Both the Reconfigure Key sent from the server to the client and the HMAC in subsequent Reconfigure messages are carried as the Authentication information in an Authentication option. The format of the Authentication information is defined in the following section.

The Reconfigure Key protocol is used (initiated by the server) only if the client and server are not using any other authentication protocol and the client and server have negotiated to use Reconfigure messages.

#### **[22.4.1.](#) Use of the Authentication Option in the Reconfigure Key Authentication Protocol**

The following fields are set in an Authentication option for the Reconfigure Key Authentication Protocol:

protocol    3

algorithm   1

RDM        0

The format of the Authentication information for the Reconfigure Key Authentication Protocol is:



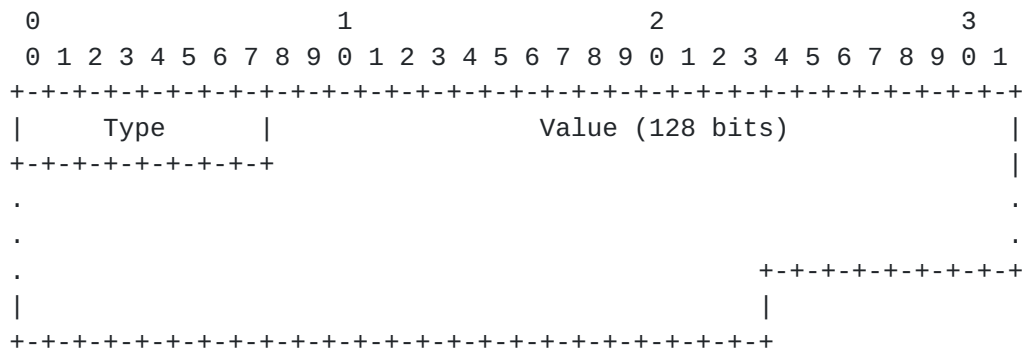


Figure 9: RKAP Authentication Information

Type	Type of data in Value field carried in this option:
1	Reconfigure Key value (used in Reply message).
2	HMAC-MD5 digest of the message (used in Reconfigure message).

Value	Data as defined by the Type field.
-------	------------------------------------

22.4.2. Server considerations for Reconfigure Key protocol

The server selects a Reconfigure Key for a client during the Request/Reply, Solicit/Reply or Information-request/Reply message exchange. The server records the Reconfigure Key and transmits that key to the client in an Authentication option in the Reply message.

The Reconfigure Key is 128 bits long, and MUST be a cryptographically strong random or pseudo-random number that cannot easily be predicted.

To provide authentication for a Reconfigure message, the server selects a replay detection value according to the RDM selected by the server, and computes an HMAC-MD5 of the Reconfigure message using the Reconfigure Key for the client. The server computes the HMAC-MD5 over the entire DHCP Reconfigure message, including the Authentication option; the HMAC-MD5 field in the Authentication option is set to zero for the HMAC-MD5 computation. The server includes the HMAC-MD5 in the authentication information field in an Authentication option included in the Reconfigure message sent to the client.



### **[22.4.3.](#) Client considerations for Reconfigure Key protocol**

The client will receive a Reconfigure Key from the server in the initial Reply message from the server. The client records the Reconfigure Key for use in authenticating subsequent Reconfigure messages.

To authenticate a Reconfigure message, the client computes an HMAC-MD5 over the DHCP Reconfigure message, using the Reconfigure Key received from the server. If this computed HMAC-MD5 matches the value in the Authentication option, the client accepts the Reconfigure message.

## **[23.](#) DHCP Options**

Options are used to carry additional information and parameters in DHCP messages. Every option shares a common base format, as described in [Section 23.1](#). All values in options are represented in network byte order.

This document describes the DHCP options defined as part of the base DHCP specification. Other options may be defined in the future in separate documents. See [[RFC7227](#)] for guidelines regarding new options definition.

Unless otherwise noted, each option may appear only in the options area of a DHCP message and may appear only once. If an option does appear multiple times, each instance is considered separate and the data areas of the options MUST NOT be concatenated or otherwise combined.

Options that are allowed to appear only once are called singleton options. The only non-singleton options defined in this document are IA\_NA (see [Section 23.4](#)), IA\_TA (see [Section 23.5](#)), and IA\_PD (see [Section 23.21](#)) options. Also, IAAddress (see [Section 23.6](#)) and IAPrefix (see [Section 23.22](#)) may appear in their respective IA options more than once.

### **[23.1.](#) Format of DHCP Options**

The format of DHCP options is:





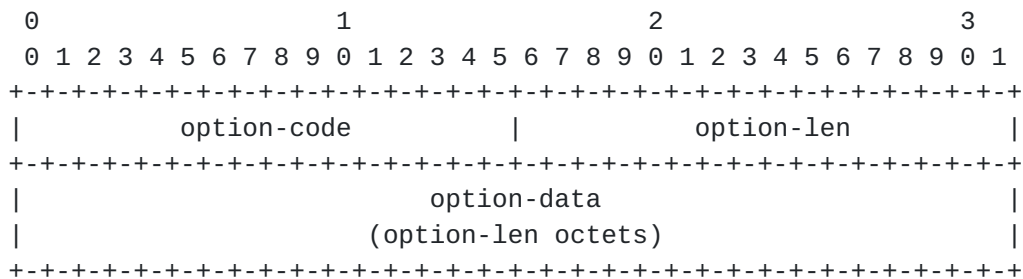


Figure 10: Option Format

option-code	An unsigned integer identifying the specific option type carried in this option.
option-len	An unsigned integer giving the length of the option-data field in this option in octets.
option-data	The data for the option; the format of this data depends on the definition of the option.

DHCPv6 options are scoped by using encapsulation. Some options apply generally to the client, some are specific to an IA, and some are specific to the addresses within an IA. These latter two cases are discussed in [Section 23.4](#) and [Section 23.6](#).

### 23.2. Client Identifier Option

The Client Identifier option is used to carry a DUID (see [Section 10](#)) identifying a client between a client and a server. The format of the Client Identifier option is:

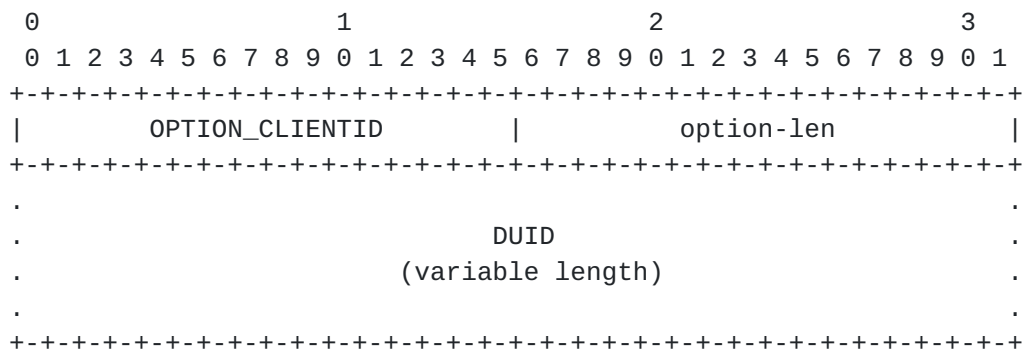


Figure 11: Client Identifier Option Format

```
option-code          OPTION_CLIENTID (1).
```



option-len                Length of DUID in octets.

DUID                     The DUID for the client.

**23.3. Server Identifier Option**

The Server Identifier option is used to carry a DUID (see [Section 10](#)) identifying a server between a client and a server. The format of the Server Identifier option is:

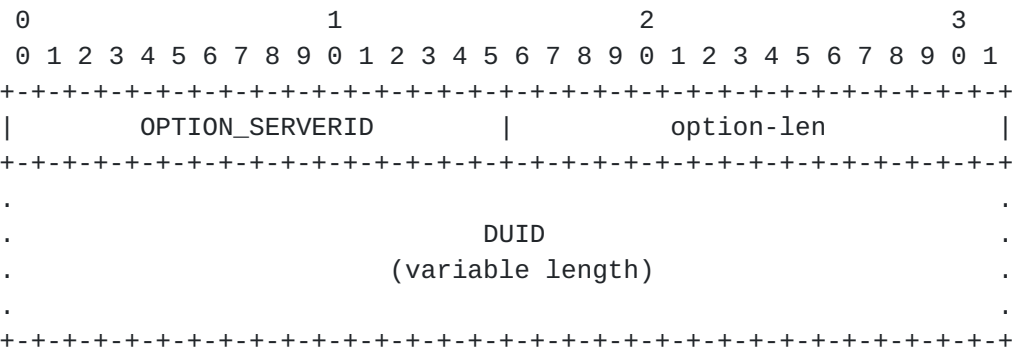


Figure 12: Server Identifier Option Format

option-code              OPTION\_SERVERID (2).

option-len               Length of DUID in octets.

DUID                     The DUID for the server.

**23.4. Identity Association for Non-temporary Addresses Option**

The Identity Association for Non-temporary Addresses option (IA\_NA option) is used to carry an IA\_NA, the parameters associated with the IA\_NA, and the non-temporary addresses associated with the IA\_NA.

Addresses appearing in an IA\_NA option are not temporary addresses (see [Section 23.5](#)).

The format of the IA\_NA option is:



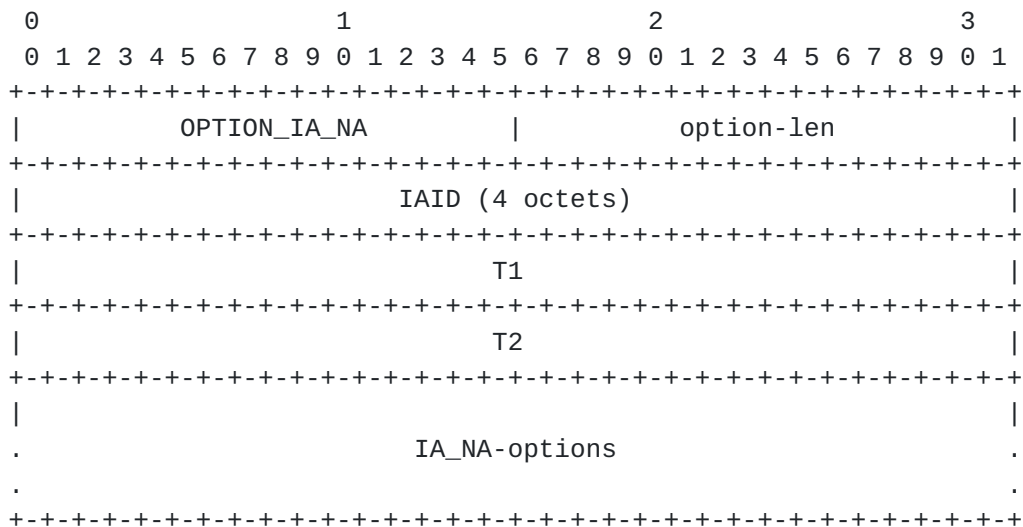


Figure 13: Identity Association for Non-temporary Addresses Option Format

option-code	OPTION_IA_NA (3).
option-len	12 + length of IA_NA-options field.
IAID	The unique identifier for this IA_NA; the IAID must be unique among the identifiers for all of this client's IA_NAs. The number space for IA_NA IAIDs is separate from the number space for IA_TA IAIDs.
T1	The time at which the client contacts the server from which the addresses in the IA_NA were obtained to extend the lifetimes of the addresses assigned to the IA_NA; T1 is a time duration relative to the current time expressed in units of seconds.
T2	The time at which the client contacts any available server to extend the lifetimes of the addresses assigned to the IA_NA; T2 is a time duration relative to the current time expressed in units of seconds.
IA_NA-options	Options associated with this IA_NA.

The IA\_NA-options field encapsulates those options that are specific to this IA\_NA. For example, all of the IA Address Options carrying



the addresses associated with this IA\_NA are in the IA\_NA-options field.

Each IA\_NA carries one "set" of non-temporary addresses; that is, at most one address from each prefix assigned to the link to which the client is attached.

An IA\_NA option may only appear in the options area of a DHCP message. A DHCP message may contain multiple IA\_NA options.

The status of any operations involving this IA\_NA is indicated in a Status Code option in the IA\_NA-options field.

Note that an IA\_NA has no explicit "lifetime" or "lease length" of its own. When the valid lifetimes of all of the addresses in an IA\_NA have expired, the IA\_NA can be considered as having expired. T1 and T2 are included to give servers explicit control over when a client recontacts the server about a specific IA\_NA.

In a message sent by a client to a server, the T1 and T2 fields SHOULD be set to 0. The server MUST ignore any values in these fields in messages received from a client.

In a message sent by a server to a client, the client MUST use the values in the T1 and T2 fields for the T1 and T2 parameters, unless those values in those fields are 0. The values in the T1 and T2 fields are the number of seconds until T1 and T2.

The server selects the T1 and T2 times to allow the client to extend the lifetimes of any addresses in the IA\_NA before the lifetimes expire, even if the server is unavailable for some short period of time. Recommended values for T1 and T2 are .5 and .8 times the shortest preferred lifetime of the addresses in the IA that the server is willing to extend, respectively. If the "shortest" preferred lifetime is 0xffffffff ("infinity"), the recommended T1 and T2 values are also 0xffffffff. If the time at which the addresses in an IA\_NA are to be renewed is to be left to the discretion of the client, the server sets T1 and T2 to 0. The client MUST follow the rules defined in [Section 14.2](#).

If a server receives an IA\_NA with T1 greater than T2, and both T1 and T2 are greater than 0, the server ignores the invalid values of T1 and T2 and processes the IA\_NA as though the client had set T1 and T2 to 0.

If a client receives an IA\_NA with T1 greater than T2, and both T1 and T2 are greater than 0, the client discards the IA\_NA option and





processes the remainder of the message as though the server had not included the invalid IA\_NA option.

Care should be taken in setting T1 or T2 to 0xffffffff ("infinity"). A client will never attempt to extend the lifetimes of any addresses in an IA with T1 set to 0xffffffff. A client will never attempt to use a Rebind message to locate a different server to extend the lifetimes of any addresses in an IA with T2 set to 0xffffffff.

This option MAY appear in a Confirm message if the lifetimes on the non-temporary addresses in the associated IA have not expired.

### 23.5. Identity Association for Temporary Addresses Option

The Identity Association for the Temporary Addresses (IA\_TA) option is used to carry an IA\_TA, the parameters associated with the IA\_TA and the addresses associated with the IA\_TA. All of the addresses in this option are used by the client as temporary addresses, as defined in [RFC4941]. The format of the IA\_TA option is:

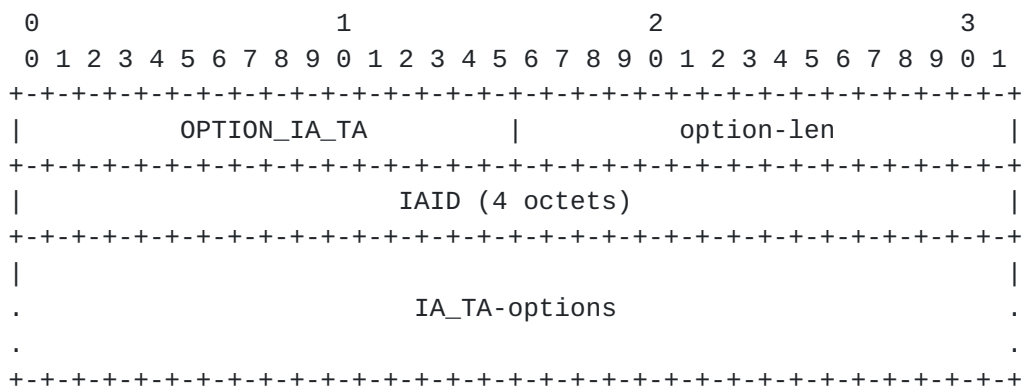


Figure 14: Identity Association for Temporary Addresses Option Format

option-code	OPTION_IA_TA (4).
option-len	4 + length of IA_TA-options field.
IAID	The unique identifier for this IA_TA; the IAID must be unique among the identifiers for all of this client's IA_TAs. The number space for IA_TA IAIDs is separate from the number space for IA_NA IAIDs.
IA_TA-options	Options associated with this IA_TA.



The IA\_TA-Options field encapsulates those options that are specific to this IA\_TA. For example, all of the IA Address Options carrying the addresses associated with this IA\_TA are in the IA\_TA-options field.

Each IA\_TA carries one "set" of temporary addresses.

An IA\_TA option may only appear in the options area of a DHCP message. A DHCP message may contain multiple IA\_TA options.

The status of any operations involving this IA\_TA is indicated in a Status Code option in the IA\_TA-options field.

Note that an IA has no explicit "lifetime" or "lease length" of its own. When the valid lifetimes of all of the addresses in an IA\_TA have expired, the IA can be considered as having expired.

An IA\_TA option does not include values for T1 and T2. A client MAY request that the lifetimes on temporary addresses be extended by including the addresses in a IA\_TA option sent in a Renew or Rebind message to a server. For example, a client would request an extension on the lifetime of a temporary address to allow an application to continue to use an established TCP connection.

The client obtains new temporary addresses by sending an IA\_TA option with a new IAID to a server. Requesting new temporary addresses from the server is the equivalent of generating new temporary addresses as described in [\[RFC4941\]](#). The server will generate new temporary addresses and return them to the client. The client should request new temporary addresses before the lifetimes on the previously assigned addresses expire.

A server MUST return the same set of temporary address for the same IA\_TA (as identified by the IAID) as long as those addresses are still valid. After the lifetimes of the addresses in an IA\_TA have expired, the IAID may be reused to identify a new IA\_TA with new temporary addresses.

This option MAY appear in a Confirm message if the lifetimes on the temporary addresses in the associated IA have not expired.

### **[23.6.](#) IA Address Option**

The IA Address option is used to specify IPv6 addresses associated with an IA\_NA or an IA\_TA. The IA Address option must be encapsulated in the Options field of an IA\_NA or IA\_TA option. The Options fields of the IA\_NA or IA\_TA option encapsulates those options that are specific to this address.



The format of the IA Address option is:

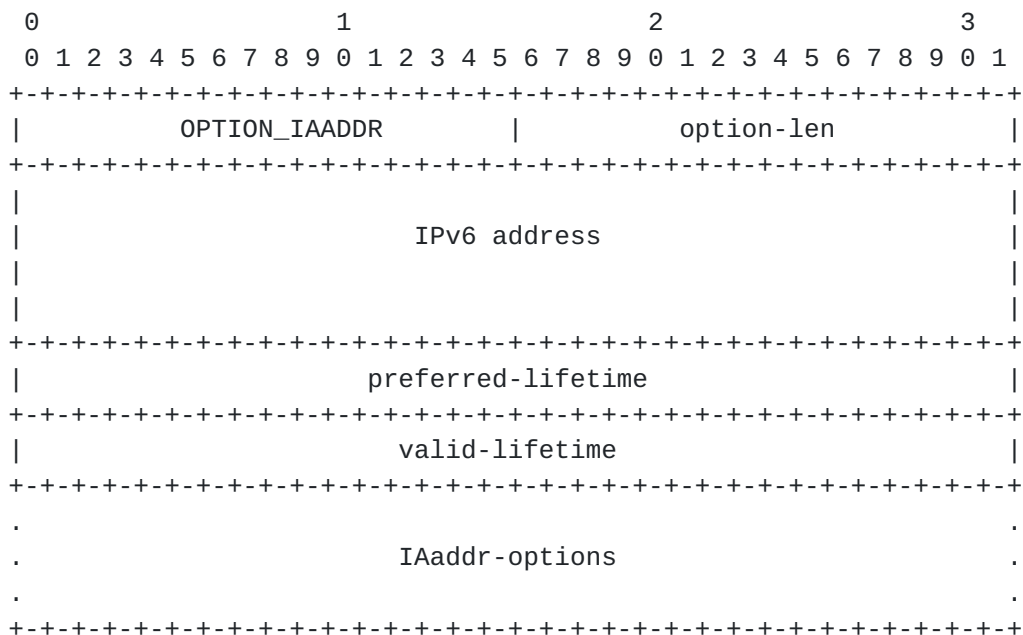


Figure 15: IA Address Option Format

option-code	OPTION_IAADDR (5).
option-len	24 + length of IAAddr-options field.
IPv6 address	An IPv6 address.
preferred-lifetime	The preferred lifetime for the IPv6 address in the option, expressed in units of seconds.
valid-lifetime	The valid lifetime for the IPv6 address in the option, expressed in units of seconds.
IAAddr-options	Options associated with this address.

In a message sent by a client to a server, the preferred and valid lifetime fields SHOULD be set to 0. The server MUST ignore any received values.

The client SHOULD NOT send the IA Address option with unspecified address (::).

In a message sent by a server to a client, the client MUST use the values in the preferred and valid lifetime fields for the preferred



and valid lifetimes. The values in the preferred and valid lifetimes are the number of seconds remaining in each lifetime.

A client discards any addresses for which the preferred lifetime is greater than the valid lifetime. A server ignores the lifetimes set by the client if the preferred lifetime is greater than the valid lifetime and ignores the values for T1 and T2 set by the client if those values are greater than the preferred lifetime.

Care should be taken in setting the valid lifetime of an address to 0xffffffff ("infinity"), which amounts to a permanent assignment of an address to a client.

More than one IA Address Option can appear in an IA\_NA option or an IA\_TA option.

The status of any operations involving this IA Address is indicated in a Status Code option in the IAAddr-options field, as specified in [Section 23.13](#).

### 23.7. Option Request Option

The Option Request option is used to identify a list of options in a message between a client and a server. The format of the Option Request option is:

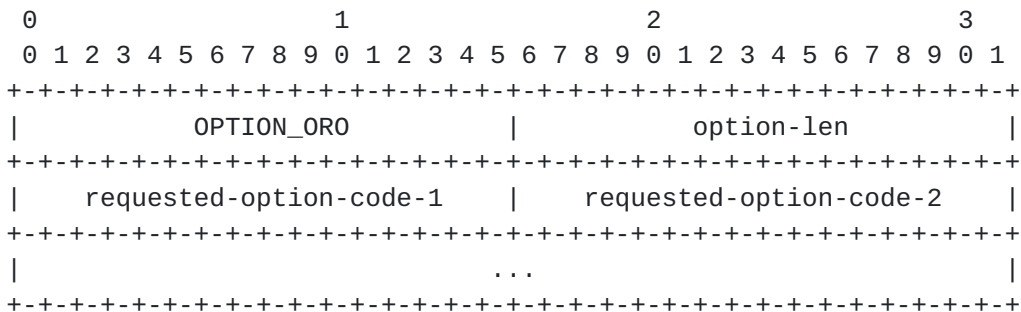


Figure 16: Option Request Option Format

option-code            OPTION\_ORO (6).

option-len            2 \* number of requested options.

requested-option-code-n    The option code for an option requested by the client.

A client MAY include an Option Request option in a Solicit, Request, Renew, Rebind, Confirm or Information-request message to inform the





server about options the client wants the server to send to the client. A server MAY include an Option Request option in a Reconfigure message to indicate which options the client should request from the server. If there is a need to request encapsulated options, top-level Option Request option MUST be used for that purpose. There is no need request IAADDR or IAPREFIX.

### [23.8.](#) Preference Option

The Preference option is sent by a server to a client to affect the selection of a server by the client.

The format of the Preference option is:

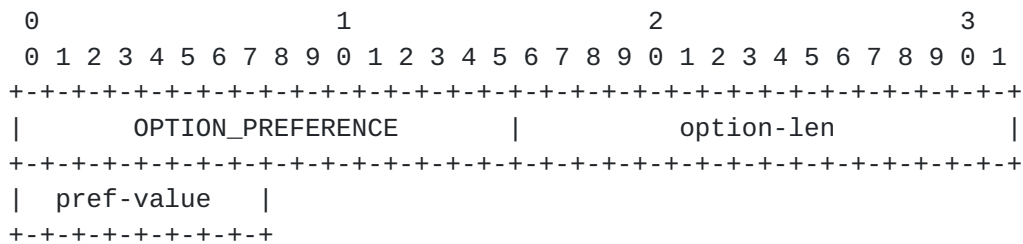


Figure 17: Preference Option Format

option-code	OPTION_PREFERENCE (7).
option-len	1.
pref-value	The preference value for the server in this message.

A server MAY include a Preference option in an Advertise message to control the selection of a server by the client. See [Section 18.1.3](#) for the use of the Preference option by the client and the interpretation of Preference option data value.

### [23.9.](#) Elapsed Time Option



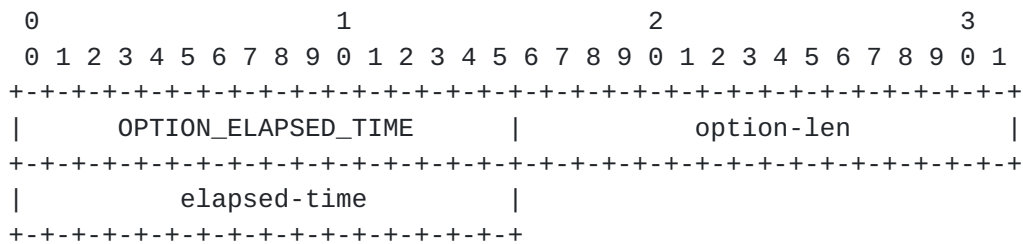


Figure 18: Elapsed Time Option Format

option-code	OPTION_ELAPSED_TIME (8).
option-len	2.
elapsed-time	The amount of time since the client began its current DHCP transaction. This time is expressed in hundredths of a second ( $10^{-2}$ seconds).

A client MUST include an Elapsed Time option in messages to indicate how long the client has been trying to complete a DHCP message exchange. The elapsed time is measured from the time at which the client sent the first message in the message exchange, and the elapsed-time field is set to 0 in the first message in the message exchange. Servers and Relay Agents use the data value in this option as input to policy controlling how a server responds to a client message. For example, the elapsed time option allows a secondary DHCP server to respond to a request when a primary server has not answered in a reasonable time. The elapsed time value is an unsigned, 16 bit integer. The client uses the value 0xffff to represent any elapsed time values greater than the largest time value that can be represented in the Elapsed Time option.

### **23.10. Relay Message Option**

The Relay Message option carries a DHCP message in a Relay-forward or Relay-reply message.

The format of the Relay Message option is:



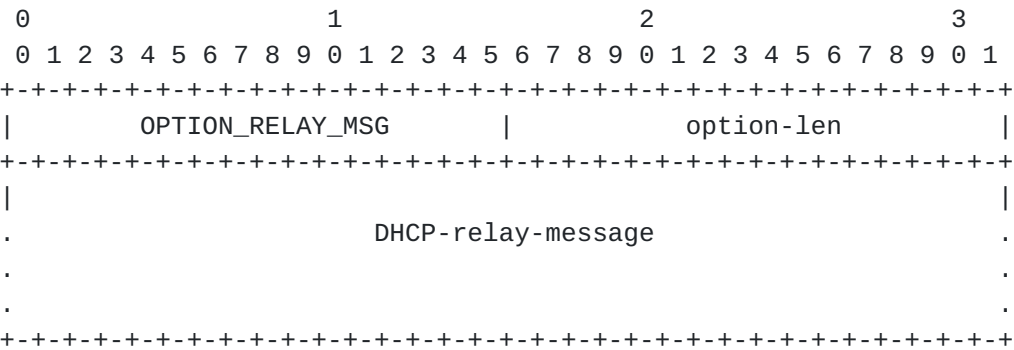


Figure 19: Relay Message Option Format

option-code	OPTION_RELAY_MSG (9)
option-len	Length of DHCP-relay-message
DHCP-relay-message	In a Relay-forward message, the received message, relayed verbatim to the next relay agent or server; in a Relay-reply message, the message to be copied and relayed to the relay agent or client whose address is in the peer-address field of the Relay-reply message

23.11. Authentication Option

The Authentication option carries authentication information to authenticate the identity and contents of DHCP messages. The use of the Authentication option is described in [Section 22](#). The format of the Authentication option is:



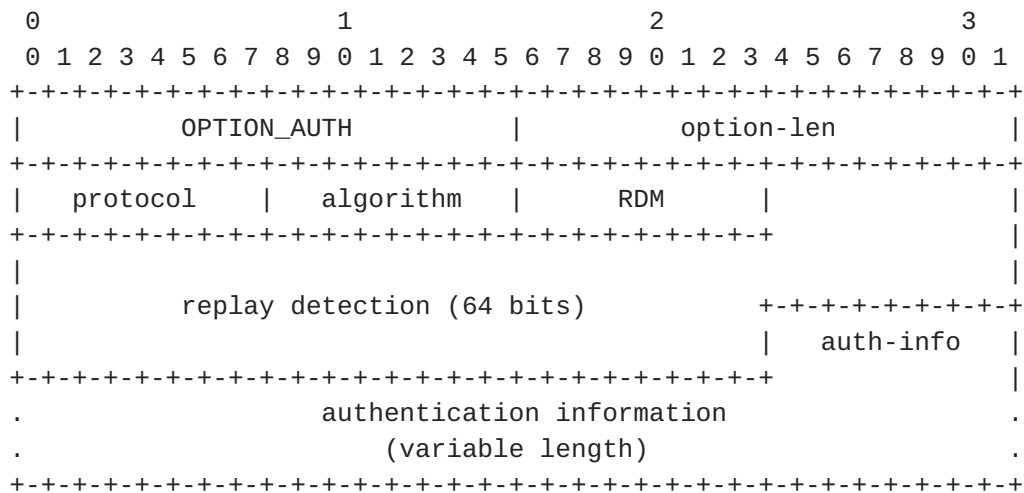


Figure 20: Authentication Option Format

option-code	OPTION_AUTH (11).
option-len	11 + length of authentication information field.
protocol	The authentication protocol used in this authentication option.
algorithm	The algorithm used in the authentication protocol.
RDM	The replay detection method used in this authentication option.
Replay detection	The replay detection information for the RDM.
authentication information	The authentication information, as specified by the protocol and algorithm used in this authentication option.

### 23.12. Server Unicast Option

The server sends this option to a client to indicate to the client that it is allowed to unicast messages to the server. The format of the Server Unicast option is:





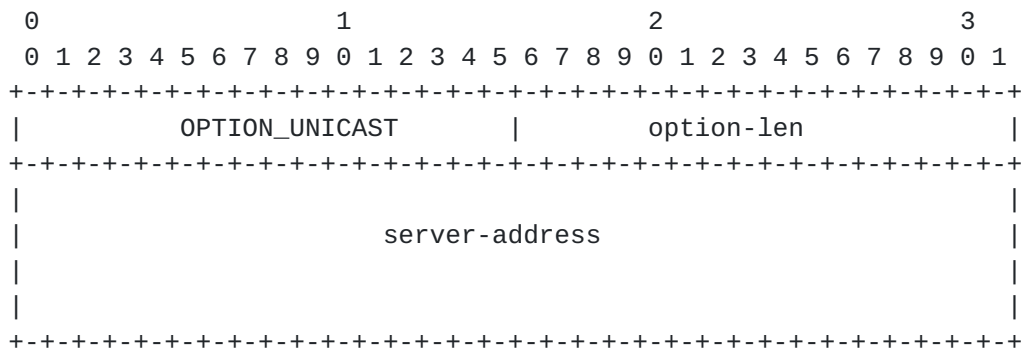


Figure 21: Server Unicast Option Format

option-code	OPTION_UNICAST (12).
option-len	16.
server-address	The IP address to which the client should send messages delivered using unicast.

The server specifies the IPv6 address to which the client is to send unicast messages in the server-address field. When a client receives this option, where permissible and appropriate, the client sends messages directly to the server using the IPv6 address specified in the server-address field of the option.

When the server sends a Unicast option to the client, some messages from the client will not be relayed by Relay Agents, and will not include Relay Agent options from the Relay Agents. Therefore, a server should only send a Unicast option to a client when Relay Agents are not sending Relay Agent options. A DHCP server rejects any messages sent inappropriately using unicast to ensure that messages are relayed by Relay Agents when Relay Agent options are in use.

Details about when the client may send messages to the server using unicast are in [Section 19](#).

### **23.13. Status Code Option**

This option returns a status indication related to the DHCP message or option in which it appears. The format of the Status Code option is:





Figure 22: Status Code Option Format

option-code	OPTION_STATUS_CODE (13).
option-len	2 + length of status-message.
status-code	The numeric code for the status encoded in this option.
status-message	A UTF-8 encoded text string suitable for display to an end user, which MUST NOT be null-terminated.

A Status Code option may appear in the options field of a DHCP message and/or in the options field of another option. If the Status Code option does not appear in a message in which the option could appear, the status of the message is assumed to be Success.

The status-codes values previously defined by [RFC3315] and [RFC3633] are:



Name	Code	Description
Success	0	Success.
UnspecFail	1	Failure, reason unspecified; this status code is sent by either a client or a server to indicate a failure not explicitly specified in this document.
NoAddrsAvail	2	Server has no addresses available to assign to the IA(s).
NoBinding	3	Client record (binding) unavailable.
NotOnLink	4	The prefix for the address is not appropriate for the link to which the client is attached.
UseMulticast	5	Sent by a server to a client to force the client to send messages to the server using the All_DHCP_Relay_Agents_and_Servers address.
NoPrefixAvail	6	Delegating router has no prefixes available to assign to the IAPD(s).

### 23.14. Rapid Commit Option

The Rapid Commit option is used to signal the use of the two message exchange for address assignment. The format of the Rapid Commit option is:

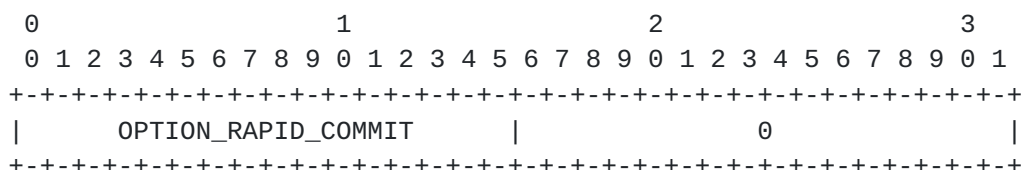


Figure 23: Rapid Commit Option Format

option-code           OPTION\_RAPID\_COMMIT (14).

option-len            0.

A client MAY include this option in a Solicit message if the client is prepared to perform the Solicit-Reply message exchange described in [Section 18.1.1](#).

A server MUST include this option in a Reply message sent in response to a Solicit message when completing the Solicit-Reply message exchange.



## DISCUSSION:

Each server that responds with a Reply to a Solicit that includes a Rapid Commit option will commit the assigned addresses in the Reply message to the client, and will not receive any confirmation that the client has received the Reply message. Therefore, if more than one server responds to a Solicit that includes a Rapid Commit option, some servers will commit addresses that are not actually used by the client.

The problem of unused addresses can be minimized, for example, by designing the DHCP service so that only one server responds to the Solicit or by using relatively short lifetimes for assigned addresses, or the DHCP client initiatively releases unused addresses using the Release message.

**23.15. User Class Option**

The User Class option is used by a client to identify the type or category of user or applications it represents.

The format of the User Class option is:

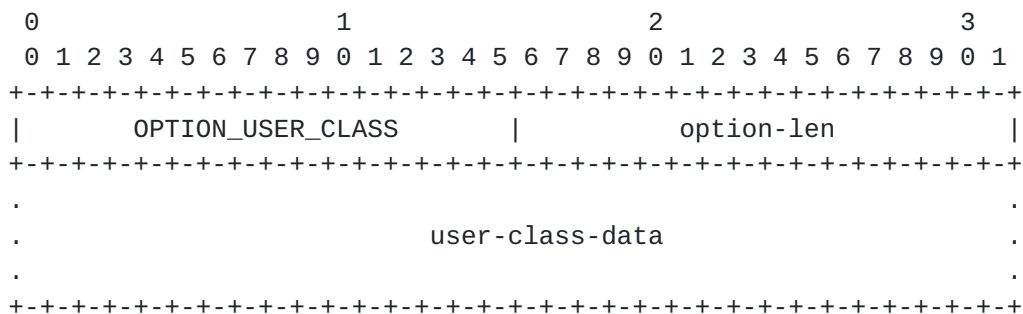


Figure 24: User Class Option Format

option-code            `OPTION_USER_CLASS` (15).

option-len            Length of user class data field.

user-class-data        The user classes carried by the client.

The information contained in the data area of this option is contained in one or more opaque fields that represent the user class or classes of which the client is a member. A server selects configuration information for the client based on the classes identified in this option. For example, the User Class option can be used to configure all clients of people in the accounting department





with a different printer than clients of people in the marketing department. The user class information carried in this option **MUST** be configurable on the client.

The data area of the user class option **MUST** contain one or more instances of user class data. Each instance of the user class data is formatted as follows:

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      user-class-len      |      opaque-data      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 25: User Class Data Format

The user-class-len is two octets long and specifies the length of the opaque user class data in network byte order.

A server interprets the classes identified in this option according to its configuration to select the appropriate configuration information for the client. A server may use only those user classes that it is configured to interpret in selecting configuration information for a client and ignore any other user classes. In response to a message containing a User Class option, a server includes a User Class option containing those classes that were successfully interpreted by the server, so that the client can be informed of the classes interpreted by the server.

### **[23.16.](#) Vendor Class Option**

This option is used by a client to identify the vendor that manufactured the hardware on which the client is running. The information contained in the data area of this option is contained in one or more opaque fields that identify details of the hardware configuration. The format of the Vendor Class option is:



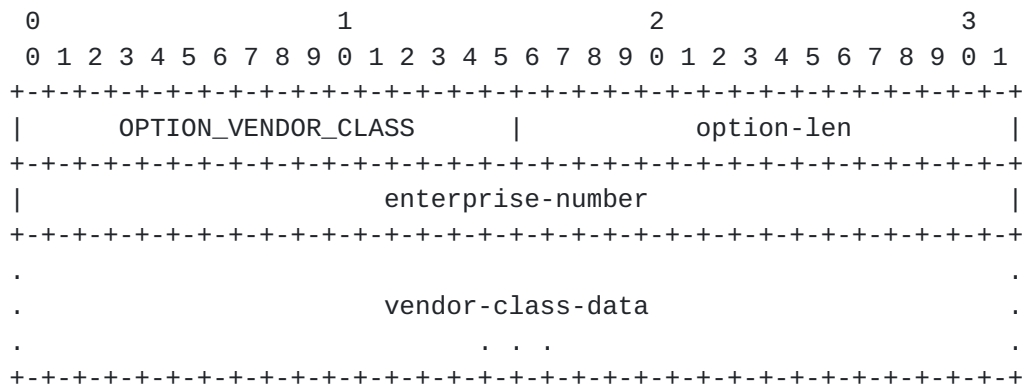


Figure 26: Vendor Class Option Format

option-code	OPTION_VENDOR_CLASS (16).
option-len	4 + length of vendor class data field.
enterprise-number	The vendor's registered Enterprise Number as registered with IANA [ <a href="#">IANA-PEN</a> ].
vendor-class-data	The hardware configuration of the host on which the client is running.

The vendor-class-data is composed of a series of separate items, each of which describes some characteristic of the client's hardware configuration. Examples of vendor-class-data instances might include the version of the operating system the client is running or the amount of memory installed on the client.

Each instance of the vendor-class-data is formatted as follows:

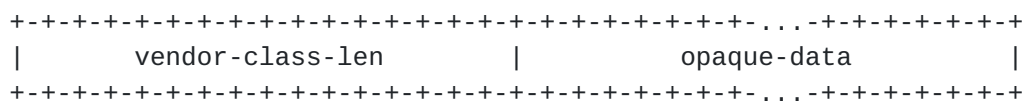


Figure 27: Vendor Class Data Format

The vendor-class-len is two octets long and specifies the length of the opaque vendor class data in network byte order.

Servers and clients MUST NOT include more than one instance of OPTION\_VENDOR\_CLASS with the same Enterprise Number. Each instance of OPTION\_VENDOR\_CLASS can carry multiple sub-options.



**23.17. Vendor-specific Information Option**

This option is used by clients and servers to exchange vendor-specific information.

The format of the Vendor-specific Information option is:

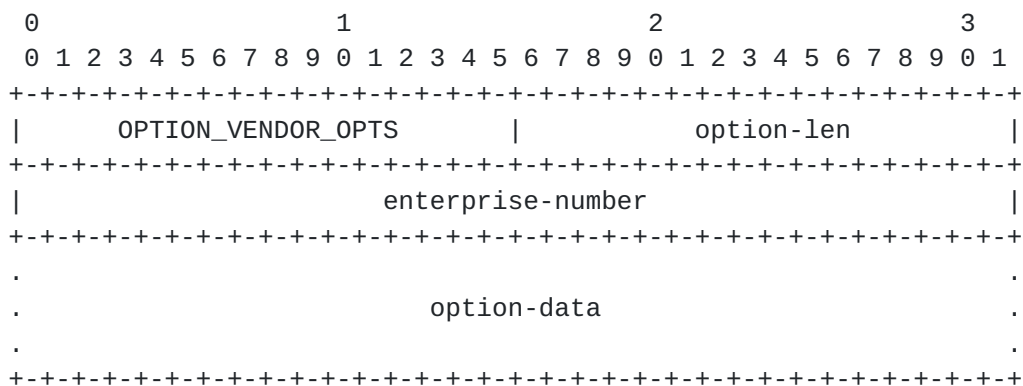


Figure 28: Vendor-specific Information Option Format

option-code	OPTION_VENDOR_OPTS (17).
option-len	4 + length of option-data field.
enterprise-number	The vendor's registered Enterprise Number as registered with IANA [ <a href="#">IANA-PEN</a> ].
option-data	An opaque object, interpreted by vendor-specific code on the clients and servers.

The definition of the information carried in this option is vendor specific. The vendor is indicated in the enterprise-number field. Use of vendor-specific information allows enhanced operation, utilizing additional features in a vendor's DHCP implementation. A DHCP client that does not receive requested vendor-specific information will still configure the host device's IPv6 stack to be functional.

The encapsulated vendor-specific options field MUST be encoded as a sequence of code/length/value fields of identical format to the DHCP options field. The option codes are defined by the vendor identified in the enterprise-number field and are not managed by IANA. Each of the encapsulated options is formatted as follows:



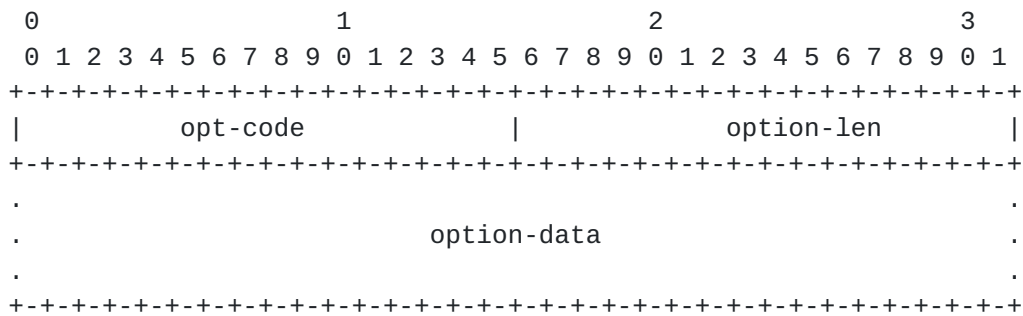


Figure 29: Vendor-specific Options Format

opt-code	The code for the encapsulated option.
option-len	An unsigned integer giving the length of the option-data field in this encapsulated option in octets.
option-data	The data area for the encapsulated option.

Multiple instances of the Vendor-specific Information option may appear in a DHCP message. Each instance of the option is interpreted according to the option codes defined by the vendor identified by the Enterprise Number in that option. Servers and clients **MUST NOT** send more than one instance of Vendor-specific Information option with the same Enterprise Number. Each instance of Vendor-specific Information option **MAY** contain multiple encapsulated options.

A client that is interested in receiving a Vendor-specific Information Option:

- **MUST** specify the Vendor-specific Information Option in an Option Request Option.
- **MAY** specify an associated Vendor Class Option.
- **MAY** specify the Vendor-specific Information Option with any data.

Servers only return the Vendor-specific Information Options if specified in Option Request Options from clients and:

- **MAY** use the Enterprise Numbers in the associated Vendor Class Options to restrict the set of Enterprise Numbers in the Vendor-specific Information Options returned.
- **MAY** return all configured Vendor-specific Information Options.





- MAY use other information in the packet or in its configuration to determine which set of Enterprise Numbers in the Vendor-specific Information Options to return.

### 23.18. Interface-Id Option

The relay agent MAY send the Interface-id option to identify the interface on which the client message was received. If a relay agent receives a Relay-reply message with an Interface-id option, the relay agent relays the message to the client through the interface identified by the option.

The format of the Interface ID option is:

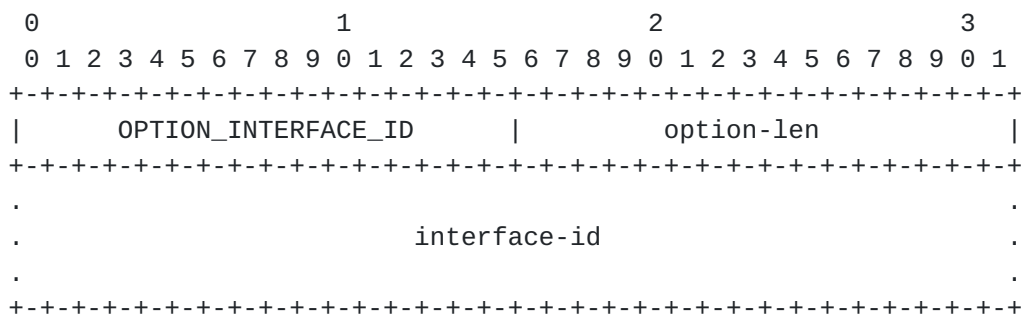


Figure 30: Interface-ID Option Format

option-code	OPTION_INTERFACE_ID (18).
option-len	Length of interface-id field.
interface-id	An opaque value of arbitrary length generated by the relay agent to identify one of the relay agent's interfaces.

The server MUST copy the Interface-Id option from the Relay-forward message into the Relay-reply message the server sends to the relay agent in response to the Relay-forward message. This option MUST NOT appear in any message except a Relay-forward or Relay-reply message.

Servers MAY use the Interface-ID for parameter assignment policies. The Interface-ID SHOULD be considered an opaque value, with policies based on exact match only; that is, the Interface-ID SHOULD NOT be internally parsed by the server. The Interface-ID value for an interface SHOULD be stable and remain unchanged, for example, after the relay agent is restarted; if the Interface-ID changes, a server will not be able to use it reliably in parameter assignment policies.



**23.19. Reconfigure Message Option**

A server includes a Reconfigure Message option in a Reconfigure message to indicate to the client whether the client responds with a Renew message, a Rebind message, or an Information-request message. The format of this option is:

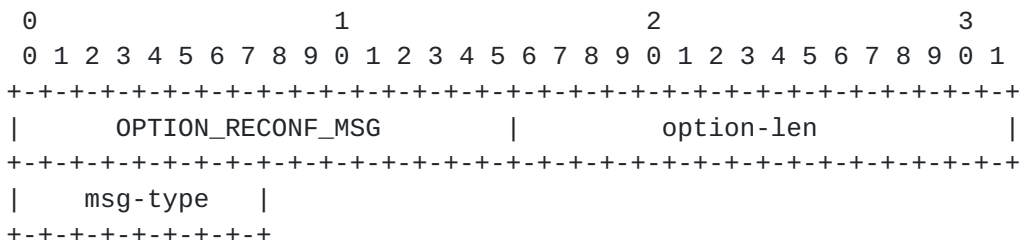


Figure 31: Reconfigure Message Option Format

option-code	OPTION_RECONF_MSG (19).
option-len	1.
msg-type	5 for Renew message, 6 for Rebind, 11 for Information-request message.

The Reconfigure Message option can only appear in a Reconfigure message.

**23.20. Reconfigure Accept Option**

A client uses the Reconfigure Accept option to announce to the server whether the client is willing to accept Reconfigure messages, and a server uses this option to tell the client whether or not to accept Reconfigure messages. The default behavior, in the absence of this option, means unwillingness to accept Reconfigure messages, or instruction not to accept Reconfigure messages, for the client and server messages, respectively. The following figure gives the format of the Reconfigure Accept option:

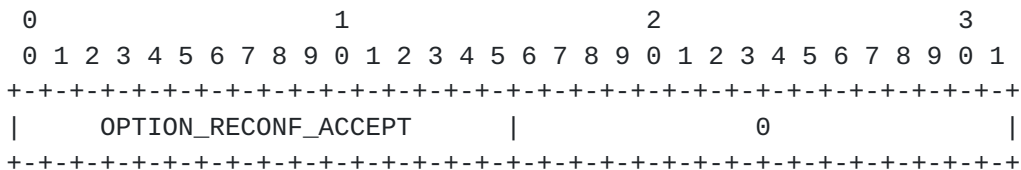


Figure 32: Reconfigure Accept Option Format



option-code           OPTION\_RECONF\_ACCEPT (20).

option-len            0.

### **23.21. Identity Association for Prefix Delegation Option**

The IA\_PD option is used to carry a prefix delegation identity association, the parameters associated with the IA\_PD and the prefixes associated with it.

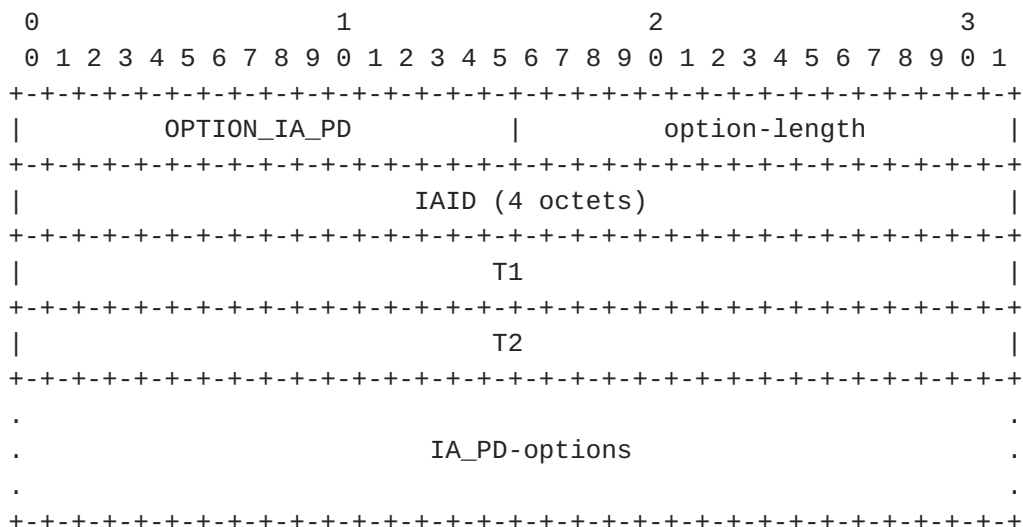


Figure 33: Identity Association for Prefix Delegation Option Format

option-code           OPTION\_IA\_PD (25).

option-length        12 + length of IA\_PD-options field.

IAID                 The unique identifier for this IA\_PD; the IAID must be unique among the identifiers for all of this requesting router's IA\_PD's.

T1                   The time at which the requesting router should contact the delegating router from which the prefixes in the IA\_PD were obtained to extend the lifetimes of the prefixes delegated to the IA\_PD; T1 is a time duration relative to the current time expressed in units of seconds.

T2                   The time at which the requesting router should contact any available delegating router to extend the lifetimes of the



prefixes assigned to the IA\_PD; T2 is a time duration relative to the current time expressed in units of seconds.

IA\_PD-options            Options associated with this IA\_PD.

The IA\_PD-options field encapsulates those options that are specific to this IA\_PD. For example, all of the IA\_PD Prefix Options carrying the prefixes associated with this IA\_PD are in the IA\_PD-options field.

An IA\_PD option may only appear in the options area of a DHCP message. A DHCP message may contain multiple IA\_PD options.

The status of any operations involving this IA\_PD is indicated in a Status Code option in the IA\_PD-options field.

Note that an IA\_PD has no explicit "lifetime" or "lease length" of its own. When the valid lifetimes of all of the prefixes in a IA\_PD have expired, the IA\_PD can be considered as having expired. T1 and T2 are included to give delegating routers explicit control over when a requesting router should contact the delegating router about a specific IA\_PD.

In a message sent by a requesting router to a delegating router, the T1 and T2 fields SHOULD be set to 0. The delegating router MUST ignore any values in these fields in messages received from a requesting router.

In a message sent by a delegating router to a requesting router, the delegating router MUST use the values in the T1 and T2 fields for the T1 and T2 parameters, unless those values in those fields are 0. The values in the T1 and T2 fields are the number of seconds until T1 and T2.

The delegating router selects the T1 and T2 times to allow the requesting router to extend the lifetimes of any prefixes in the IA\_PD before the lifetimes expire, even if the delegating router is unavailable for some short period of time. Recommended values for T1 and T2 are .5 and .8 times the shortest preferred lifetime of the prefixes in the IA\_PD that the delegating router is willing to extend, respectively. If the time at which the prefixes in an IA\_PD are to be renewed is to be left to the discretion of the requesting router, the delegating router sets T1 and T2 to 0. The requesting router MUST follow the rules defined in [Section 14.2](#).

If a delegating router receives an IA\_PD with T1 greater than T2, and both T1 and T2 are greater than 0, the delegating router ignores the





invalid values of T1 and T2 and processes the IA\_PD as though the requesting router had set T1 and T2 to 0.

If a requesting router receives an IA\_PD with T1 greater than T2, and both T1 and T2 are greater than 0, the requesting router discards the IA\_PD option and processes the remainder of the message as though the requesting router had not included the IA\_PD option.

### [23.22.](#) IA Prefix Option

The IA\_PD Prefix option is used to specify IPv6 address prefixes associated with an IA\_PD. The IA\_PD Prefix option must be encapsulated in the IA\_PD-options field of an IA\_PD option.

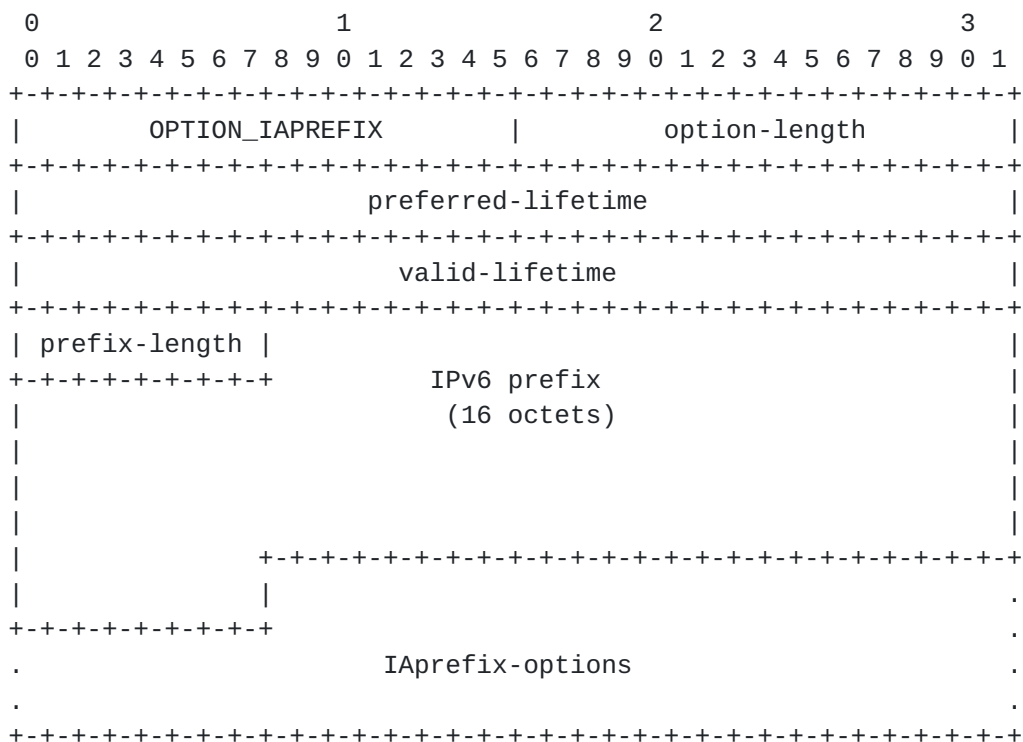


Figure 34: IA Prefix Option Format

option-code	OPTION_IAPREFIX (26).
option-length	25 + length of IAprefix-options field.
preferred-lifetime	The recommended preferred lifetime for the IPv6 prefix in the option, expressed in units of seconds. A value of 0xFFFFFFFF represents infinity.



valid-lifetime	The valid lifetime for the IPv6 prefix in the option, expressed in units of seconds. A value of 0xFFFFFFFF represents infinity.
prefix-length	Length for this prefix in bits.
IPv6-prefix	An IPv6 prefix.
IAprefix-options	Options associated with this prefix.

In a message sent by a requesting router to a delegating router, the preferred and valid lifetime fields SHOULD be set to 0. The server MUST ignore any received values in these lifetime fields.

A requesting router may set the IPv6 prefix field to zero and a given value in the prefix-length field to indicate a preference for the size of the prefix to be delegated.

In a message sent by a delegating router the preferred and valid lifetimes should be set to the values of AdvPreferredLifetime and AdvValidLifetime as specified in [section 6.2.1](#), "Router Configuration Variables" of [[RFC2461](#)], unless administratively configured.

A requesting router discards any prefixes for which the preferred lifetime is greater than the valid lifetime. A delegating router ignores the lifetimes set by the requesting router if the preferred lifetime is greater than the valid lifetime and ignores the values for T1 and T2 set by the requesting router if those values are greater than the preferred lifetime.

The values in the preferred and valid lifetimes are the number of seconds remaining for each lifetime.

An IA\_PD Prefix option may appear only in an IA\_PD option. More than one IA\_PD Prefix Option can appear in a single IA\_PD option.

The status of any operations involving this IA\_PD Prefix option is indicated in a Status Code option in the IAprefix-options field.

### **[23.23.](#) SOL\_MAX\_RT Option**

A DHCP server sends the SOL\_MAX\_RT option to a client to override the default value of SOL\_MAX\_RT. The value of SOL\_MAX\_RT in the option replaces the default value defined in [Section 6.5](#). One use for the SOL\_MAX\_RT option is to set a longer value for SOL\_MAX\_RT, which reduces the Solicit traffic from a client that has not received a response to its Solicit messages.



The format of the SOL\_MAX\_RT option is:

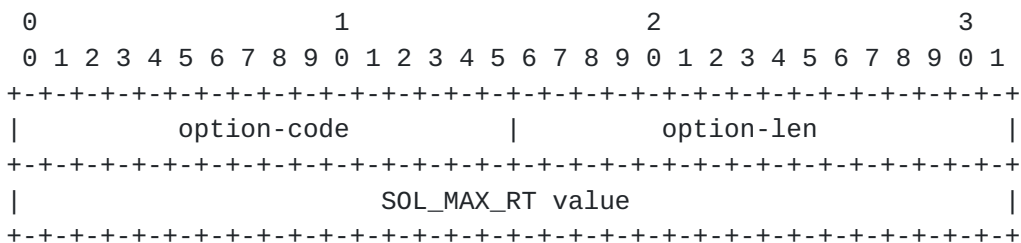


Figure 35: SOL\_MAX\_RT Option Format

option-code	OPTION_SOL_MAX_RT (82).
option-len	4.
SOL_MAX_RT value	Overriding value for SOL_MAX_RT in seconds; MUST be in range: 60 <= "value" <= 86400 (1 day).

A DHCP client MUST include the SOL\_MAX\_RT option code in any Option Request option (see [Section 23.7](#)) it sends.

The DHCP server MAY include the SOL\_MAX\_RT option in any response it sends to a client that has included the SOL\_MAX\_RT option code in an Option Request option. The SOL\_MAX\_RT option is sent in the main body of the message to client, not as an encapsulated option in, e.g., an IA\_NA, IA\_TA, or IA\_PD option.

A DHCP client MUST ignore any SOL\_MAX\_RT option values that are less than 60 or more than 86400.

If a DHCP client receives a message containing a SOL\_MAX\_RT option that has a valid value for SOL\_MAX\_RT, the client MUST set its internal SOL\_MAX\_RT parameter to the value contained in the SOL\_MAX\_RT option. This value of SOL\_MAX\_RT is then used by the retransmission mechanism defined in [Section 15](#) and [Section 18.1.2](#).

Updated SOL\_MAX\_RT value applies only to the network interface on which the client received SOL\_MAX\_RT option.

#### **23.24. INF\_MAX\_RT Option**

A DHCP server sends the INF\_MAX\_RT option to a client to override the default value of INF\_MAX\_RT. The value of INF\_MAX\_RT in the option replaces the default value defined in [Section 6.5](#). One use for the INF\_MAX\_RT option is to set a longer value for INF\_MAX\_RT, which



reduces the Information-request traffic from a client that has not received a response to its Information-request messages.

The format of the INF\_MAX\_RT option is:

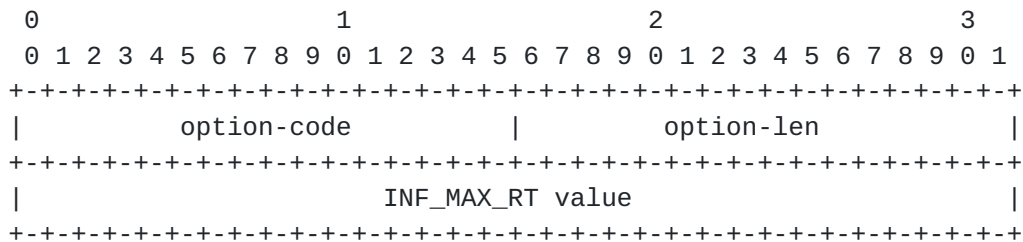


Figure 36: INF\_MAX\_RT Option Format

option-code	OPTION_INF_MAX_RT (83).
option-len	4.
SOL_MAX_RT value	Overriding value for INF_MAX_RT in seconds; MUST be in range: 60 <= "value" <= 86400 (1 day).

```
option-len      4.
```

SOL_MAX_RT value	Overriding value for INF_MAX_RT in seconds; MUST be in range: 60 <= "value" <= 86400 (1 day).
------------------	---

A DHCP client MUST include the INF\_MAX\_RT option code in any Option Request option (see [Section 23.7](#)) it sends.

The DHCP server MAY include the INF\_MAX\_RT option in any response it sends to a client that has included the INF\_MAX\_RT option code in an Option Request option. The INF\_MAX\_RT option is sent in the main body of the message to client, not as an encapsulated option in, e.g., an IA\_NA, IA\_TA, or IA\_PD option.

A DHCP client MUST ignore any INF\_MAX\_RT option values that are less than 60 or more than 86400.

If a DHCP client receives a message containing an `INF_MAX_RT` option that has a valid value for `INF_MAX_RT`, the client MUST set its internal `INF_MAX_RT` parameter to the value contained in the `INF_MAX_RT` option. This value of `INF_MAX_RT` is then used by the retransmission mechanism defined in [Section 15](#) and [Section 19.1.5](#).

Updated `INF_MAX_RT` value applies only to the network interface on which the client received `INF_MAX_RT` option.





## **24. Security Considerations**

The threat to DHCP is inherently an insider threat (assuming a properly configured network where DHCPv6 ports are blocked on the perimeter gateways of the enterprise). Regardless of the gateway configuration, however, the potential attacks by insiders and outsiders are the same.

Use of manually configured preshared keys for IPsec between relay agents and servers does not defend against replayed DHCP messages. Replayed messages can represent a DOS attack through exhaustion of processing resources, but not through mis-configuration or exhaustion of other resources such as assignable addresses.

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 misconfiguration of the client that causes all network communication from the client to fail.

A malicious DHCP server might cause a client to set its SOL\_MAX\_RT and INF\_MAX\_RT parameters to an unreasonably high value with the SOL\_MAX\_RT and INF\_MAX\_RT options, which may cause an undue delay in a client completing its DHCP protocol transaction in the case no other valid response is received. Assuming the client also receives a response from a valid DHCP server, large values for SOL\_MAX\_RT and INF\_MAX\_RT will not have any effect.

There is another threat to DHCP clients from mistakenly or accidentally configured DHCP servers that answer DHCP client requests with unintentionally incorrect configuration parameters.

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. Note that although a client sends its response (Renew or Information-request message) through a relay agent and, therefore, that response will only be received by servers to which DHCP messages are relayed, a malicious server could send a Reconfigure message to a client, followed (after an appropriate delay) by a Reply message that would be accepted by the client. Thus, a malicious server that is not on the network path between the client and the server may still be able to mount a Reconfigure attack on a client. The use of transaction IDs that are cryptographically sound and cannot easily be predicted



will also reduce the probability that such an attack will be successful.

The threat specific to a DHCP server is an invalid client masquerading as a valid client. The motivation for this may be for theft of service, or to circumvent auditing for any number of nefarious purposes.

The threat common to both the client and the server is the resource "denial of service" (DoS) attack. These attacks typically involve the exhaustion of available addresses, or the exhaustion of CPU or network bandwidth, and are present anytime there is a shared resource.

In the case where relay agents add additional options to Relay Forward messages, the messages exchanged between relay agents and servers may be used to mount a "man in the middle" or denial of service attack.

This threat model does not consider the privacy of the contents of DHCP messages to be important. DHCP is not used to exchange authentication or configuration information that must be kept secret from other networks nodes.

DHCP authentication provides for authentication of the identity of DHCP clients and servers, and for the integrity of messages delivered between DHCP clients and servers. DHCP authentication does not provide any privacy for the contents of DHCP messages.

Because of the opportunity for attack through the Reconfigure message, a DHCP client MUST discard any Reconfigure message that does not include authentication or that does not pass the validation process for the authentication protocol.

The Reconfigure Key protocol described in [Section 22.4](#) provides protection against the use of a Reconfigure message by a malicious DHCP server to mount a denial of service or man-in-the-middle attack on a client. This protocol can be compromised by an attacker that can intercept the initial message in which the DHCP server sends the key to the client.

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 22.1](#). The use of manual configuration and installation of static keys are acceptable in this instance because relay agents and the server will belong to the same administrative domain and the relay agents will require other specific configuration



(for example, configuration of the DHCP server address) as well as the IPsec configuration.

A rogue delegating router can issue bogus prefixes to a requesting router. This may cause denial of service due to unreachability.

A malicious requesting router may be able to mount a denial of service attack by repeated requests for delegated prefixes that exhaust the delegating router's available prefixes.

To guard against attacks through prefix delegation, requesting routers and delegating routers SHOULD use DHCP authentication as described in [Section 22](#). For point to point links, where one trusts that there is no man in the middle, or one trusts layer two authentication, DHCP authentication or IPsec may not be necessary. Because a requesting router and delegating routers must each have at least one assigned IPv6 address, the routers may be able to use IPsec for authentication of DHCPv6 messages. The details of using IPsec for DHCPv6 are under development.

Networks configured with delegated prefixes should be configured to preclude intentional or inadvertent inappropriate advertisement of these prefixes.

## **[25.](#) IANA Considerations**

This document does not define any new DHCPv6 name spaces or definitions.

IANA is requested to update the <http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xhtml> page to add a reference to this document for definitions previously created by [[RFC3315](#)], [[RFC3633](#)], and [[RFC7083](#)].

## **[26.](#) Obsoleted mechanisms**

This specification of the DHCPv6 is mostly a corrected and cleaned up version of the original spec [[RFC3315](#)] along with numerous additions from later RFCs. However, there is a small number of mechanisms that didn't get much traction, were not widely deployed, underspecified or had other operational issues. Those mechanisms are now considered deprecated. Legacy implementations MAY support it, but implementations conformant to this document MUST NOT rely on them.

The following mechanism are now obsolete:



Delayed Authentication. This mechanism was underspecified and had significant operational burden. As a result, after 10 years its adoption was extremely limited at best.

Lifetime hints sent by a client. Client used to be allowed to send lifetime values as hints. This mechanism was not widely implemented and there were known misimplementations that sent remaining lifetimes rather than total lifetimes. That in turn was sometimes misunderstood by the servers as a request for ever decreasing lease lifetimes, which caused issues when values started approaching zero.

## **27. Acknowledgments**

The following people are authors of the original [RFC 3315](#): Ralph Droms, Jim Bound, Bernie Volz, Ted Lemon, Charles Perkins, and Mike Carney. The following people are authors of the original [RFC 3633](#): Ole Troan and Ralph Droms. This document is merely a refinement of their work and would not be possible without their original work.

A number of additional people have contributed to identifying issues with [RFC 3315](#) and [RFC 3633](#) and proposed resolutions to these issues as reflected in this document (in no particular order): Ole Troan, Robert Marks, Leaf Yeh, Tim Winters, Michelle Cotton, Pablo Armando, John Brzozowski, Suresh Krishnan, Hideshi Enokihara, Alexandru Petrescu, Yukiyo Akisada, Tatuya Jinmei, Fred Templin. With special thanks to Ralph Droms for answering many questions related to the original [RFC 3315](#) work.

The following acknowledgements are from the original [RFC 3315](#) and [RFC 3633](#):

Thanks to the DHC Working Group and the members of the IETF for their time and input into the specification. In particular, thanks also for the consistent input, ideas, and review by (in alphabetical order) Bernard Aboba, Bill Arbaugh, Thirumalesh Bhat, Steve Bellovin, A. K. Vijayabhaskar, Brian Carpenter, Matt Crawford, Steve Deering, Francis Dupont, Dave Forster, Brian Haberman, Richard Hussong, Tatuya Jinmei, Kim Kinnear, Fredrik Lindholm, Tony Lindstrom, Josh Littlefield, Gerald Maguire, Jack McCann, Shin Miyakawa, Thomas Narten, Erik Nordmark, Jarno Rajahalme, Yakov Rekhter, Pekka Savola, Mark Stapp, Matt Thomas, Sue Thomson, Tatuya Jinmei, Bernie Volz, Trevor Warwick, Phil Wells and Toshi Yamasaki.

Thanks to Steve Deering and Bob Hinden, who have consistently taken the time to discuss the more complex parts of the IPv6 specifications.





And, thanks to Steve Deering for pointing out at IETF 51 in London that the DHCPv6 specification has the highest revision number of any Internet Draft.

## **[28.](#) References**

### **[28.1.](#) Normative References**

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, [RFC 768](#), DOI 10.17487/RFC0768, August 1980, <<http://www.rfc-editor.org/info/rfc768>>.
- [RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, [RFC 826](#), DOI 10.17487/RFC0826, November 1982, <<http://www.rfc-editor.org/info/rfc826>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, [RFC 1035](#), DOI 10.17487/RFC1035, November 1987, <<http://www.rfc-editor.org/info/rfc1035>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), DOI 10.17487/RFC2131, March 1997, <<http://www.rfc-editor.org/info/rfc2131>>.
- [RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", [RFC 2132](#), DOI 10.17487/RFC2132, March 1997, <<http://www.rfc-editor.org/info/rfc2132>>.
- [RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", [RFC 2136](#), DOI 10.17487/RFC2136, April 1997, <<http://www.rfc-editor.org/info/rfc2136>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), DOI 10.17487/RFC2460, December 1998, <<http://www.rfc-editor.org/info/rfc2460>>.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", [RFC 2464](#), DOI 10.17487/RFC2464, December 1998, <<http://www.rfc-editor.org/info/rfc2464>>.



- [RFC2526] Johnson, D. and S. Deering, "Reserved IPv6 Subnet Anycast Addresses", [RFC 2526](#), DOI 10.17487/RFC2526, March 1999, <<http://www.rfc-editor.org/info/rfc2526>>.
- [RFC3118] Droms, R. and W. Arbaugh., Ed., "Authentication for DHCP Messages", [RFC 3118](#), DOI 10.17487/RFC3118, June 2001, <<http://www.rfc-editor.org/info/rfc3118>>.
- [RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3646](#), DOI 10.17487/RFC3646, December 2003, <<http://www.rfc-editor.org/info/rfc3646>>.
- [RFC4075] Kalusivalingam, V., "Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6", [RFC 4075](#), DOI 10.17487/RFC4075, May 2005, <<http://www.rfc-editor.org/info/rfc4075>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 4291](#), DOI 10.17487/RFC4291, February 2006, <<http://www.rfc-editor.org/info/rfc4291>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", [RFC 4301](#), DOI 10.17487/RFC4301, December 2005, <<http://www.rfc-editor.org/info/rfc4301>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#), DOI 10.17487/RFC4861, September 2007, <<http://www.rfc-editor.org/info/rfc4861>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", [RFC 4862](#), DOI 10.17487/RFC4862, September 2007, <<http://www.rfc-editor.org/info/rfc4862>>.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", [RFC 4941](#), DOI 10.17487/RFC4941, September 2007, <<http://www.rfc-editor.org/info/rfc4941>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", [RFC 5905](#), DOI 10.17487/RFC5905, June 2010, <<http://www.rfc-editor.org/info/rfc5905>>.



- [RFC6221] Miles, D., Ed., Ooghe, S., Dec, W., Krishnan, S., and A. Kavanagh, "Lightweight DHCPv6 Relay Agent", [RFC 6221](#), DOI 10.17487/RFC6221, May 2011, <<http://www.rfc-editor.org/info/rfc6221>>.
- [RFC6355] Narten, T. and J. Johnson, "Definition of the UUID-Based DHCPv6 Unique Identifier (DUID-UUID)", [RFC 6355](#), DOI 10.17487/RFC6355, August 2011, <<http://www.rfc-editor.org/info/rfc6355>>.
- [RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", [RFC 6724](#), DOI 10.17487/RFC6724, September 2012, <<http://www.rfc-editor.org/info/rfc6724>>.
- [RFC7083] Droms, R., "Modification to Default Values of SOL\_MAX\_RT and INF\_MAX\_RT", [RFC 7083](#), DOI 10.17487/RFC7083, November 2013, <<http://www.rfc-editor.org/info/rfc7083>>.
- [RFC7227] Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and S. Krishnan, "Guidelines for Creating New DHCPv6 Options", [BCP 187](#), [RFC 7227](#), DOI 10.17487/RFC7227, May 2014, <<http://www.rfc-editor.org/info/rfc7227>>.
- [RFC7283] Cui, Y., Sun, Q., and T. Lemon, "Handling Unknown DHCPv6 Messages", [RFC 7283](#), DOI 10.17487/RFC7283, July 2014, <<http://www.rfc-editor.org/info/rfc7283>>.

## 28.2. Informative References

- [I-D.cui-dhc-dhcpv6-prefix-length-hint-issue]  
Cui, Y., Li, T., and C. Liu, "DHCPv6 Prefix Length Hint Issues", [draft-cui-dhc-dhcpv6-prefix-length-hint-issue-01](#) (work in progress), October 2015.
- [I-D.ietf-dhc-topo-conf]  
Lemon, T. and T. Mrugalski, "Customizing DHCP Configuration on the Basis of Network Topology", [draft-ietf-dhc-topo-conf-05](#) (work in progress), July 2015.
- [IANA-PEN]  
IANA, "Private Enterprise Numbers registry", <http://www.iana.org/assignments/enterprise-numbers>.
- [RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", [RFC 2461](#), DOI 10.17487/RFC2461, December 1998, <<http://www.rfc-editor.org/info/rfc2461>>.



- [RFC2462] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", [RFC 2462](#), DOI 10.17487/RFC2462, December 1998, <<http://www.rfc-editor.org/info/rfc2462>>.
- [RFC3041] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", [RFC 3041](#), DOI 10.17487/RFC3041, January 2001, <<http://www.rfc-editor.org/info/rfc3041>>.
- [RFC3162] Aboba, B., Zorn, G., and D. Mitton, "RADIUS and IPv6", [RFC 3162](#), DOI 10.17487/RFC3162, August 2001, <<http://www.rfc-editor.org/info/rfc3162>>.
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3315](#), DOI 10.17487/RFC3315, July 2003, <<http://www.rfc-editor.org/info/rfc3315>>.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", [RFC 3633](#), DOI 10.17487/RFC3633, December 2003, <<http://www.rfc-editor.org/info/rfc3633>>.
- [RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", [RFC 3736](#), DOI 10.17487/RFC3736, April 2004, <<http://www.rfc-editor.org/info/rfc3736>>.
- [RFC3769] Miyakawa, S. and R. Droms, "Requirements for IPv6 Prefix Delegation", [RFC 3769](#), DOI 10.17487/RFC3769, June 2004, <<http://www.rfc-editor.org/info/rfc3769>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", [RFC 4193](#), DOI 10.17487/RFC4193, October 2005, <<http://www.rfc-editor.org/info/rfc4193>>.
- [RFC7084] Singh, H., Beebe, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", [RFC 7084](#), DOI 10.17487/RFC7084, November 2013, <<http://www.rfc-editor.org/info/rfc7084>>.
- [RFC7341] Sun, Q., Cui, Y., Siodelski, M., Krishnan, S., and I. Farrer, "DHCPv4-over-DHCPv6 (DHCP 4o6) Transport", [RFC 7341](#), DOI 10.17487/RFC7341, August 2014, <<http://www.rfc-editor.org/info/rfc7341>>.





[RFC7550] Troan, O., Volz, B., and M. Siodelski, "Issues and Recommendations with Multiple Stateful DHCPv6 Options", [RFC 7550](#), DOI 10.17487/RFC7550, May 2015, <<http://www.rfc-editor.org/info/rfc7550>>.

#### **Appendix A.** Changes since [RFC3315](#)

1. Incorporated [RFC3315](#) errata (ids: 294, 1373, 2928, 1815, 3577, 2509, 295).
2. Partially incorporated [RFC3315](#) errata id 2472 (place other IA options if NoAddrsAvail is sent in Advertise).
3. Clarified [section 21.4.1 of RFC3315](#) by defining length of "key ID" field and specifying that 'DHCP realm' is Domain Name encoded as per [section 8 of RFC3315](#). Ticket #43.
4. Added DUID-UUID and reference to [RFC6355](#). Ticket #54.
5. Specified a minimum length for the DUID in section "9.1. DUID Contents". Ticket #39.
6. Removed the use of term "sub-options" from section "19.1.1. Creation and Transmission of Reconfigure Messages". Ticket #40.
7. Added text to [section 22.6](#) "IA Address Option" about the usage of unspecified address to express the client hints for Preferred and Valid lifetimes. Ticket #45.
8. Updated text in 21.4.2 of [RFC3315](#) ("Message Validation") as suggested in section 3.1 of [draft-ietf-dhc-dhcpv6-clarify-auth-01](#). Ticket #87.
9. Merged [RFC7083](#), "Modification to Default Values of SOL\_MAX\_RT and INF\_MAX\_RT", into this document. Ticket #51.
10. Incorporated [RFC3315](#) errata (id 2471), into [section 17.1.3](#). Ticket #25.
11. Added text that relay agents MUST NOT modify the relayed message to [section 20.1.2](#). Ticket #57.
12. Modified the text in [section 21.4.4.5](#), Receiving Reply Messages, to remove special treatment of a Reply validation failure (client ignores message). Ticket #89.
13. [Appendix C](#) updated: Authentication option is no longer allowed in Relay-forward and Relay-reply messages, ORO is no longer



- allowed in Confirm, Release and Decline messages; Preference option is no longer allowed in Reply messages (only in Advertise). Ticket #10.
14. Removed "silently" from several instances of "silently ignores" or "silently" discards. It is up to software vendor if and how to log such events (debug log message, event log, message pop-up etc.). Ticket #50.
  15. Clarified that: there should be no more than one instance of Vendor Class option with a given Enterprise Number; that one instance of Vendor Class can contain multiple encapsulated options; the same applies to Vendor Specific Information option. Ticket #22.
  16. Clarified relay agent definition. Ticket #12.
  17. Changed REL\_MAX\_RC and DEC\_MAX\_RC defaults from 5 to 4 and added retry to parameter description. Ticket #84.
  18. Clarify handling process for Vendor-specific Information Option and Vendor Class Option. Ticket #20.
  19. Replace "monotonic" with "strictly monotonic" in [Section 21.3](#). Ticket #11.
  20. Incorporate everything of [RFC 6644](#), except for Security Considerations Section, which has already been covered in a more abstracted way. Ticket #55 & #56.
  21. Clarify the server behavior process when a client violates Delayed Authentication Protocol, in [Section 21.4](#). Ticket #90.
  22. Updated titles of sections [19.4.2](#) and 19.4.4. to include Rebind messages.
  23. Applied many of the review comments from a review done by Fred Templin in August 2006. Ticket #14.
  24. Reworded the first paragraph of [Section 15](#) to relax the "SHOULD" requirement to drop the messages which contain the options not expected in the current message. Ticket #17.
  25. Changed WG to DHC, added keywords
  26. Loosened requirements for DUID-EN, so that DUID type can be used for virtual machines. Ticket #16.



27. Clarified that IA may contain other resources than just address. Ticket #93.
28. Clarified that most options are singletons (i.e. can appear only once). Ticket #83.
29. Merged sections [1](#) (Ticket #96), 2 (Ticket #97), 3 (Ticket #98), 4 (Ticket #99), 6 (Ticket #101), 8 (Ticket #103), 9 (Ticket #104), 10 (Ticket #105), 11 (Ticket #106), 13 (Ticket #108), 14 (Ticket #109), 15 (Ticket #110), 16 (Ticket #111), 17 (Ticket #112) and 19 (Ticket #113) from [RFC3633](#) (Prefix Delegation).
30. Clarified that encapsulated options must be requested using top level ORO (ticket #38).
31. Clarified that configuration for interface X should be requested over interface X (ticket #48).
32. CONFIRM is now an optional message (MUST send Confirm eased to SHOULD) (ticket #120).
33. Added reference to [RFC7227](#): DHCPv6 Option Guidelines (ticket #121).
34. Added new [section 5](#) providing an overview of DHCPv6 operational modes and removed two prefix delegation sections from [section 1](#). See tickets #53, #100, and #102.
35. Addressed ticket #115 - don't use DHCPv6 for DHCPv4 configuration.
36. Revised IANA Considerations based on ticket #117.
37. Updated IAID description in the terminology with the clarification that the IAID is unique among IAs of a specific type, rather than globally unique among all IAs (ticket #94).
38. Merged [Section 12](#) from [RFC3633](#) (ticket #107)
39. Clarified behavior for unknown messages ([RFC7283](#)), ticket #58.
40. Addressed tickets #123 and #126, and clarified that the client SHOULD abandon its bindings when restarts the server solicitation.
41. Clarified link-address field usage, ticket #73.



42. Clarified that retransmitting at client's discretion ( $t_1, t_2=0$ ) does not mean immediate transmission (ticket #71).
43. Merged [section 4.4. of RFC7550](#) (stateful-issues). This includes new Renew and Rebind processing by the server, i.e. a client may request allocation of new addresses/prefixes during Renew and Rebind. This addresses tickets #62 and #63.
44. Merged [section 4.1](#) and 4.2 of [RFC7550](#) (stateful-issues). This addresses tickets #59 and #60.
45. Added normative reference to [RFC7550](#) and to obsoleted list. Also added terse text of Abstract and [section 1.0](#) to indicate what this document is about.
46. Merged [section 4.6 of RFC7550](#) (stateful-issues). This addresses ticket #85.
47. Clarified that the document assumes single provisioning domain. This resolves ticket #66.
48. Removed the Delayed Auth Protocol. This resolves ticket #86.
49. Added text the text suggesting to not send all-zero address/prefix unless the client wishes to hint the lifetimes and/or prefix lengths. This resolves ticket#82.
50. Cleaned up Confirm text regarding rebooting client an stable storage.
51. Updated server processing section to clarify that server unicast check can be based on current configuration of unicast setting so server is not required to remember what it sent. This resolves ticket #130.
52. Removed references to lifetime hints from the client; only IAPREFIX prefix-length hints are now allowed. Servers will ignore any lifetime hints; clients should set lifetime to 0. This resolves ticket #148. Also, removed T1/T2 hints from clients to servers - while not explicitly discussed, seems to make sense? And, fixed some of the message processing text to include PD, not just addresses (though I think Marcin will eventually replace much of this text? Plus a few other minor edits.
53. Clarified that the server may return different addresses in the Reply than requested by the client in the Request message. Also





clarified that the server must not include addresses that it will not assign. This resolves ticket #69.

54. Introduction updated, including relation to previous RFCs (ticket #136).
55. Added section with deprecated mechanisms (ticket #149).
56. Combined text for prefix delegation and address assignment in sections [18](#) and [19](#). Also, introduced a term "lease" replacing "allocable resources". This resolves ticket #146.

#### [Appendix B](#). Changes since [RFC3633](#)

1. Incorporated [RFC3633](#) errata (ids: 248, 1880, 2468, 2469, 2470, 3736)
2. ...

#### [Appendix C](#). Appearance of Options in Message Types

The following table indicates with a "\*" the options are allowed in each DHCP message type:

	Client ID	Server ID	IA_NA IA_TA	IA_PD	Option Request	Pref	Elap. Time	Relay Msg.	Auth.	Server Unicast
Solicit	*		*	*	*		*		*	
Advert.	*	*	*	*		*			*	
Request	*	*	*	*	*		*		*	
Confirm	*		*				*		*	
Renew	*	*	*	*	*		*		*	
Rebind	*		*	*	*		*		*	
Decline	*	*	*	*			*		*	
Release	*	*	*	*			*		*	
Reply	*	*	*	*					*	*
Reconf.	*	*			*				*	
Inform.	*	(see note)			*		*		*	
R-forw.								*		
R-repl.								*		

NOTE:

Only included in Information-request messages that are sent in response to a Reconfigure (see [Section 20.4.3](#)).



	Status Code	Rap. Comm.	User Class	Vendor Class	Vendor Spec.	Inter. ID	Recon. Msg.	Recon. Accept	SOL_MAX_RT INF_MAX_RT
Solicit		*	*	*	*			*	
Advert.	*		*	*	*			*	*
Request			*	*	*			*	
Confirm			*	*	*				
Renew			*	*	*			*	
Rebind			*	*	*			*	
Decline			*	*	*				
Release			*	*	*				
Reply	*	*	*	*	*			*	*
Reconf.							*		
Inform.			*	*	*			*	
R-forw.			*	*	*	*			
R-repl.			*	*	*	*			

#### [Appendix D](#). Appearance of Options in the Options Field of DHCP Options

The following table indicates with a "\*" where options can appear in the options field of other options:

	Option Field	IA_NA/ IA_TA	IAADDR	IA_PD	IAPREFIX	Relay Forw.	Relay Reply
Client ID	*						
Server ID	*						
IA_NA/IA_TA	*						
IAADDR		*					
IA_PD	*						
IAPREFIX				*			
ORO	*						
Preference	*						
Elapsed Time	*						
Relay Message						*	*
Authentic.	*						
Server Uni.	*						
Status Code	*	*		*			
Rapid Comm.	*						
User Class	*						
Vendor Class	*						
Vendor Info.	*					*	*
Interf. ID						*	*
Reconf. MSG.	*						
Reconf. Accept	*						

Note: "Relay Forw" / "Relay Reply" options appear in the options field of the message but may only appear in these messages.



## Authors' Addresses

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

Email: [tomasz.mrugalski@gmail.com](mailto:tomasz.mrugalski@gmail.com)

Marcin Siodelski  
Internet Systems Consortium, Inc.  
950 Charter St.  
Redwood City, CA 94063  
USA

Email: [msiodelski@gmail.com](mailto:msiodelski@gmail.com)

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

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

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

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

Michael C. Richardson  
Sandelman Software Works  
470 Dawson Avenue  
Ottawa, ON K1Z 5V7  
CA

Email: [mcr+ietf@sandelman.ca](mailto:mcr+ietf@sandelman.ca)  
URI: <http://www.sandelman.ca/>



Sheng Jiang  
Huawei Technologies Co., Ltd  
Q14, Huawei Campus, No.156 Beiqing Road  
Hai-Dian District, Beijing, 100095  
P.R. China

Email: [jiangsheng@huawei.com](mailto:jiangsheng@huawei.com)

Ted Lemon  
Nominum, Inc.  
950 Charter St.  
Redwood City, CA 94043  
USA

Email: [Ted.Lemon@nominum.com](mailto:Ted.Lemon@nominum.com)



