Network Working Group
Internet-Draft
Updates: 3315,3633 (if approved)
Intended status: Standards Track
Expires: May 30, 2015

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

November 26, 2014

Issues and Recommendations with Multiple Stateful DHCPv6 Options
draft-ietf-dhc-dhcpv6-stateful-issues-09.txt

Abstract

Dynamic Host Configuration Protocol for IPv6 (DHCPv6) was not written with the expectation that additional stateful DHCPv6 options would be developed. IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6 has since shoe-horned a new option for Prefix Delegation into DHCPv6. Implementation experience of the CPE model described in RFC 7084 has shown multiple issues with the DHCPv6 protocol in supporting multiple stateful options. This document updates RFC 3315 and RFC 3633 to address the identified issues.

Table of Contents

1. Introduction

DHCPv6 [RFC3315] was not written with the expectation that additional stateful DHCPv6 options would be developed. DHCPv6 Prefix Delegation [RFC3633] has since shoe-horned a new option for Prefix Delegation into DHCPv6. Implementation experience of the CPE model described in [RFC7084] has shown issues with the DHCPv6 protocol in supporting multiple stateful option types, in particular IA_NA (non-temporary addresses) and IA_PD (delegated prefixes).

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

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

Note that while IA_TA (temporary addresses) options may be included with other IA option type requests, these generally are not renewed (there are no T1/T2 times) and have a separate life cycle from IA_NA and IA_PD option types. DHCPv6 assigned temporary addresses also have limited value when DHCPv6 is used for non-temporary address assignment, as the privacy issues identified for IPv6 stateless address assignment ([RFC4941]) do not apply to DHCPv6 assignments. Therefore, the IA_TA option type is mostly out of scope for this document.

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

2. Conventions

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

3. Terminology

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

Identity association (IA):	Throughout this document, "IA" is used to refer to the Identity Association containing addresses or prefixes assigned to a client and carried in the IA_NA or IA_PD options respectively.
IA option types:	This is used to generally mean an IA_NA and/or IA_PD option.
Stateful options:	Options that require dynamic binding state per client on the server.

4. Handling of Multiple IA Options Types

DHCPv6 was written with the assumption that the only stateful options were for assigning addresses. DHCPv6 PD describes how to extend the DHCPv6 protocol to handle prefix delegation, but [RFC3633] did not

consider how DHCP address assignment and prefix delegation could co-exist.

If a client requests multiple IA option types, but the server is configured to only offer a subset of them, the client could react in several ways. Reset the state machine and continue to send Solicit messages, create separate DHCP sessions for each IA option type and continue to Solicit for the unfulfilled IA options, or it could continue with the single session, and include the unfulfilled IA options in subsequent messages to the server.

Reseting the state machine and continuing to send Solicit messages may result in the client never completing DHCP and is generally not considered a good solution. It can also result in a request storm if the client does not appropriately rate limit its sending of Solicit messages.

Creating a separate DHCP session (separate instances of the client state machine) per IA option type, while conceptually simple, has a number of issues: multiple instances of the state machine in clients, additional DHCP protocol traffic, 'collisions' between other configuration options, divergence in that each IA option type specification specifies its 'own' version of the DHCP protocol.

This leaves a single DHCP session and state machine which is the proposed solution. Here, the client can use what it is able to obtain and can continue to request what it was previously unable to obtain while maintaining a single session and state machine.

Proposed solution: the client should keep a single session with the server and include the missing options in subsequent messages (Request, Renew, and Rebind) to the server.

4.1. Placement of Status Codes

In Reply messages IA specific status codes (i.e., NoAddrsAvail, NotOnLink, NoBinding, NoPrefixAvail) are encapsulated in the IA option. In Advertise messages the Status Code option with the NoAddrsAvail code is in the top level. This makes sense if the client is only interested in the assignment of the addresses and the failure case is fatal. However, if the client sends both IA_NA and IA_PD options in a Solicit message, it is possible that the server offers no addresses but it offers some prefixes, and the client may choose to send a Request message to obtain the offered prefixes. In this case, it is better if the Status Code option for IA specific status codes is encapsulated in the IA option to indicate that the failure occurred for the specific IA. This also makes the

NoAddrsAvail and NoPrefixAvail Status Code option placement for Advertise messages identical to Reply messages.

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

Therefore, the proposed solution is:

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

1. Advertise containing just a top-level Status Code option (of NoAddrsAvail) and no IA_NAs/IA_TAs.

2. Advertise containing the IA_NAs and/or IA_TAs with encapsulated Status Code option (of NoAddrsAvail) and no top-level Status Code option.

3. Advertise containing a top-level Status Code option (of NoAddrsAvail) and IA_NAs and/or IA_TAs with a Status Code option (of NoAddrsAvail).

Servers MUST return the Status Code option (of NoAddrsAvail) encapsulated in an IA_NA/IA_TA options and not as a top-level Status Code option (of NoAddrsAvail) when no addresses will be assigned (2 in the above list). This means that the Advertise response matches the Reply response with respect to the handling of the NoAddrsAvail status.

Replace the following paragraph in RFC 3315, section 17.2.2:

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

With:

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

4.2. Advertise Message

[RFC3315] specifies that a client must ignore an Advertise message if a server will not assign any addresses to a client. A client requesting both IA_NA and IA_PD, with a server that only offers one of them, is not supported in the current protocol specification.

Proposed solution: a client SHOULD accept Advertise messages, even when not all IA option types are being offered. And, in this case, the client SHOULD include the not offered IA option types in its Request. A client SHOULD only ignore an Advertise message when all IA options include no offered addresses or delegated prefixes. Note that ignored messages MUST still be processed for SOL_MAX_RT and INF_MAX_RT options as specified in [RFC7083].

Replace Section 17.1.3 of RFC 3315: (existing errata)

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

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

> The client MUST ignore any Advertise message that contains no addresses (IAADDR options encapsulated in IA_NA or IA_TA options) and no delegated prefixes (IAPREFIX options encapsulated in IA_PD options, see RFC 3633) with the exception that the client MUST process an included SOL_MAX_RT option (RFC 7083), MUST process an included INF_MAX_RT option (RFC 7083), and MAY display any associated status message(s) to the user.

And, replace:

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

With:

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

It is important to note that the receipt of an Advertise message without any addresses and delegated prefixes does not imply that the client should restart the Solicit retransmissions timers. Doing so would lead to a Solicit/Advertise storm.

4.3. T1/T2 Timers

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

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

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

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

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

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

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

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

4.4. Renew and Rebind Messages

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

4.4.1. Renew Message

The Renew message, as described in [RFC3315], allows a client to only renew bindings assigned via a Request message.

In a multiple IA option type model, the Renew does not support the ability for the client to renew one IA option type while requesting bindings for other IA option types that were not available when the client sent the Request.

Proposed solution: The client should continue with the IA options received, while continuing to include the other IA options in subsequent messages to the server. The client and server processing need to be modified. Note that this change makes the server's IA processing of Renew similar to the Request processing.

4.4.2. Rebind Message

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

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

The Rebind message, as described in [RFC3315] does not explicitly specify what a server should do when an IA option which contains no addresses is present.

Proposed solution: The client should continue with the IA options received and it MAY include additional IA options to request creation of additional bindings.

4.4.3. Updates to section 18.1.3 of RFC 3315

Replace Section 18.1.3 of RFC 3315 with the following text:

> To extend the valid and preferred lifetimes for the addresses associated with an IA, the client sends a Renew message to the server from which the client obtained the addresses in the IA containing an IA option for the IA. The client includes IA Address options in the IA option for the addresses associated with the IA. The server determines new lifetimes for the addresses in the IA according to the administrative configuration of the server. The server may also add new addresses to the IA. The server may remove addresses from the IA by setting the preferred and valid lifetimes of those addresses to zero.
>
> The server controls the time at which the client contacts the server to extend the lifetimes on assigned addresses through the T1 and T2 parameters assigned to an IA. However, as the client Renews/Rebinds all IAs from the server at the same time, the client MUST select a T1 and T2 time from all IA options which will guarantee that the client will send Renew/Rebind messages not later than at the T1/T2 times associated with any of the client's bindings.
>
> At time T1, the client initiates a Renew/Reply message exchange to extend the lifetimes on any addresses in the IA.
>
> If T1 or T2 had been set to 0 by the server (for an IA_NA) or there are no T1 or T2 times (for an IA_TA) in a previous Reply, the client may send a Renew or Rebind message, respectively, at the client's discretion.
>
> The client sets the "msg-type" field to RENEW. The client generates a transaction ID and inserts this value in the "transaction-id" field.
>
> The client places the identifier of the destination server in a Server Identifier option.
>
> The client MUST include a Client Identifier option to identify itself to the server. The client adds any appropriate options, including one or more IA options.
>
> The client includes an IA option with all addresses currently assigned to the IA in its Renew message. The client also includes IA options for all other bindings for which the client desires to extend the lifetimes of addresses. The client MUST only include

addresses in the IA that the client obtained from the server and are still valid (have non-zero lifetime).

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

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

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

IRT REN_TIMEOUT

MRT REN_MAX_RT

MRC 0

MRD Remaining time until T2

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

4.4.4. Updates to Section 18.1.4 of RFC 3315

Replace Section 18.1.4 of RFC 3315 with the following text:

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

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

- The client sets the "msg-type" field to REBIND.

- The client does not include the Server Identifier option in the Rebind message.

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

IRT REB_TIMEOUT

MRT REB_MAX_RT

MRC 0

MRD Remaining time until valid lifetimes of all addresses in all IAs have expired

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

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

4.4.5. Updates to Section 18.1.8 of RFC 3315

Replace Section 18.1.8 of RFC 3315 with the following text:

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

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

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

When the client receives a NotOnLink status from the server in response to a Confirm message, the client performs DHCP server solicitation, as described in section 17, and client-initiated configuration as described in section 18. If the client receives

any Reply messages that do not indicate a NotOnLink status, the client can use the addresses in the IA and ignore any messages that indicate a NotOnLink status.

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

The client SHOULD perform duplicate address detection [17] on each of the received addresses in any IAs, on which it has not performed duplicate address detection during processing of any of the previous Reply messages from the server. The client performs the duplicate address detection before using the received addresses for the traffic. If any of the addresses are found to be in use on the link, the client sends a Decline message to the server for those addresses as described in section 18.1.7.

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

- Record T1 and T2 times.

- Add any new addresses in the IA option to the IA as recorded by the client.

- Update lifetimes for any addresses in the IA option that the client already has recorded in the IA.

- Discard any addresses from the IA, as recorded by the client, that have a valid lifetime of 0 in the IA Address option.

- Leave unchanged any information about addresses the client has recorded in the IA but that were not included in the IA from the server.

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

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

If the client finds no usable addresses in any of the IAs, it may either try another server (by restarting the DHCP server discovery

process) or use the Information-request message to obtain other configuration information only.

The client uses addresses and other information from any IAs that do not contain a Status Code option with the NoAddrsAvail code. For each IA for which the client receives NoAddrsAvail status code the client has the following choices:

- The client includes the IA with no addresses in subsequent Renew and Rebind messages sent to the server, to request creation of the binding for the IA.

- Tries another server (by restarting the DHCP server discovery process).

When the client receives a Reply message in response to a Renew or Rebind message, for each IA in the original Renew or Rebind message, the client:

- Sends a Request message if the server responded with the NoBinding status code. The client places only these IA options in the Request message for which the server returned 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 the IA is not in the Reply message. However, in this case, the client MUST limit the rate at which it sends subsequent Renew/Rebind messages and limit the duration of the time during which it sends the messages.

- Otherwise accepts the information in the IA.

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

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

4.4.6. Updates to Section 18.2.3 of RFC 3315

Replace Section 18.2.3 of RFC 3315 with the following text:

When the server receives a Renew message via unicast from a client to which the server has not sent a unicast option, the server discards the Renew message and responds with a Reply message containing a Status Code option with the value UseMulticast, a

Server Identifier option containing the server's DUID, the Client Identifier option from the client message, and no other options.

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

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

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

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

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

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

- If the server is configured to create new bindings as a result of processing Renew messages, but the server will not assign any addresses to an IA, the server returns the IA option containing a Status Code option with the NoAddrsAvail status code and a status message for a user.

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

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

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

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

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

4.4.7. Updates to Section 18.2.4 of RFC 3315

Replace Section 18.2.4 of RFC 3315 with the following text:

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

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

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

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

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

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

- If the server is configured to create new bindings as a result of processing Rebind messages, but the server will not assign any addresses to an IA, the server returns the IA option containing a Status Code option with the NoAddrsAvail status code and a status message for a user.

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

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

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

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

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

The T1/T2 times set in each applicable IA option for a Reply MUST be the same values across all IAs. The server MUST determine the

T1/T2 times across all of the applicable client's bindings in the Reply. This facilitates the client being able to renew all of the bindings at the same time.

4.4.8. Updates to RFC 3633

Replace Section 12.1:

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

With:

> Each prefix has valid and preferred lifetimes whose durations are specified in the IA_PD Prefix option for that prefix. The requesting router uses Renew and Rebind messages to request the extension of the lifetimes of a delegated prefix.
>
> The requesting router MAY include IA_PD options without any prefixes, i.e. without IA Prefix option or with IPv6 prefix field of IA Prefix option set to 0, in a Renew or Rebind message to obtain bindings it desires but has been unable to obtain. The requesting router MAY set the prefix-length field of the IA Prefix option as a hint to the server.

Replace Section 12.2:

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

With:

> For the Renew or Rebind, if the IA_PD contains no prefixes, i.e. contains no IA Prefix option or the IPv6 prefix field in the IA Prefix option is set to 0, the delegating router SHOULD assign prefixes to the IA_PD according to the delegating router's explicit configuration information. In this case, when the server assigns new prefixes to the IA_PD, the server MAY use the value in the prefix-length field of the IA Prefix option as a hint for the length of the prefixes being assigned.
>
> The delegating router behaves as follows when it cannot find a binding for the requesting router's IA_PD containing prefixes:

4.5. Confirm Message

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

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

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

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

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

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

4.6. Decline Should Not Necessarily Trigger a Release

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

It is recommended that a client SHOULD NOT send a Release message for other bindings it may have received just because it sent a Decline message. The client should retain the non-conflicting bindings.

4.7. Multiple Provisioning Domains

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

One could envision a network where the DHCP servers are in multiple provisioning domains, and it may be desirable to have the DHCP client obtain different IA types from different provisioning domains. How a client detects the multiple provisioning domains and how it would interact with the multiple servers in these different domains is outside the scope of this document.

5. IANA Considerations

This specification does not require any IANA actions.

6. Security Considerations

There are no new security considerations pertaining to this document.

7. Acknowledgements

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

8. References

8.1. Normative References

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

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

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

[RFC7083] Droms, R., "Modification to Default Values of SOL_MAX_RT and INF_MAX_RT", RFC 7083, November 2013.

8.2. Informative References

[I-D.dhcwg-dhc-rfc3315bis] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., and T. Lemon, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) bis", draft-dhcwg-dhc-rfc3315bis-03 (work in progress), October 2014.

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

[RFC6059] Krishnan, S. and G. Daley, "Simple Procedures for Detecting Network Attachment in IPv6", RFC 6059, November 2010.

[RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, November 2013.

[RFC7227] Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and S. Krishnan, "Guidelines for Creating New DHCPv6 Options", BCP 187, RFC 7227, May 2014.

Authors' Addresses

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

Email: ot@cisco.com

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

Email: volz@cisco.com

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

Email: msiodelski@gmail.com