

DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 29, 2014

D. Raghuvanshi
K. Kinnear
D. Kukrety
Cisco Systems, Inc.
March 28, 2014

**DHCPv6 Active Leasequery
draft-ietf-dhc-dhcpv6-active-leasequery-01**

Abstract

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) has been extended with a Leasequery capability that allows a requestor to request information about DHCPv6 bindings. That mechanism is limited to queries for DHCPv6 binding data updates prior to the time the DHCPv6 server receives the Leasequery request. Continuous update of an external requestor with Leasequery data is sometimes desired. This document expands on the DHCPv6 Leasequery protocol, and allows for active transfer of real-time DHCPv6 binding information data via TCP. This document also extends DHCPv6 Bulk Leasequery by adding new options.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 29, 2014.

Copyright Notice

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

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

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

Table of Contents

1.	Introduction	3
2.	Terminology	3
3.	Protocol Overview	5
4.	Interaction Between Active Leasequery and Bulk Leasequery	6
5.	Extension to DHCPv6 Bulk Leasequery	7
6.	Message and Option Definitions	7
6.1.	Message Framing for TCP	7
6.2.	Messages	7
6.2.1.	ACTIVELEASEQUERY	8
6.3.	Options	8
6.3.1.	OPTION_LQ_BASE_TIME	9
6.3.2.	OPTION_LQ_START_TIME	9
6.3.3.	OPTION_LQ_END_TIME	10
6.4.	Connection and Transmission Parameters	11
7.	Information Communicated by Active Leasequery	11
8.	Requestor Behavior	12
8.1.	Connecting and General Processing	12
8.2.	Forming an Active Leasequery	13
8.3.	Processing Active Replies	14
8.3.1.	Processing Replies from a Request Containing a OPTION_LQ_START_TIME	15
8.4.	Processing Time Values in Leasequery messages	18
8.5.	Examples	18
8.5.1.	Query Failure	19
8.5.2.	Data Missing on Server	19
8.5.3.	Successful Query	19
8.6.	Closing Connections	20
9.	Server Behavior	20
9.1.	Accepting Connections	20
9.2.	Replying to an Active Leasequery	20
9.3.	Multiple or Parallel Queries	22
9.4.	Closing Connections	23
10.	Security Considerations	23
11.	IANA Considerations	24
12.	Acknowledgements	25
13.	Modification History	25
14.	References	25
14.1.	Normative References	25
14.2.	Informative References	25

Authors' Addresses	26
------------------------------	--------------------

[1.](#) Introduction

The DHCPv6 [[RFC3315](#)] protocol specifies a mechanism for the assignment of IPv6 address and configuration information to IPv6 nodes. IPv6 Prefix Delegation for DHCPv6 (PD) [[RFC3633](#)] specifies a mechanism for DHCPv6 delegation of IPv6 prefixes and related data. DHCPv6 servers maintain authoritative information including binding information for delegated IPv6 prefixes.

Requirements exist for external entities to keep up to date on the correspondence between DHCPv6 clients and their bindings. These requirements often stem from regulatory requirements placed on service providers by governmental agencies.

These entities need to keep up with the current binding activity of the DHCPv6 server. Keeping up with these binding activity is termed "active" leasequery.

The DHCPv6 Bulk Leasequery [[RFC5460](#)] capability can be used to recover useful information from a DHCPv6 server when some external entity starts up. This entity could be one which is directly involved in the DHCPv6 client - server transactions (e.g., a relay agent), or it could be an external process which needs information present in the DHCPv6 server's lease state database.

The Active Leasequery capability documented here is designed to allow an entity not directly involved in DHCPv6 client - server transactions to nevertheless keep current with the state of the DHCPv6 lease state information in real-time.

[2.](#) Terminology

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

DHCPv6 terminology is defined in [[RFC3315](#)]. Terminology specific to DHCPv6 Active Leasequery can be found below:

- o "Absolute Time"

- A 32-bit quantity containing the number of seconds since midnight January 1, 2000 UTC.

- o "Active Leasequery"

Keeping up to date in real-time (or near real-time) with DHCPv6 binding activity.

- o "Bulk Leasequery"

Requesting and receiving the existing DHCPv6 binding information in an efficient manner.

- o "binding change/update"

Any change in the DHCPv6 binding state or data stored on the DHCPv6 server related to binding. This also includes expiration or deletion of the binding.

- o "catch-up information, catch-up phase"

If a DHCPv6 Active Leasequery requestor sends `OPTION_LQ_START_TIME` option in an `ACTIVELEASEQUERY` message, the DHCPv6 server will attempt to send the requestor the information that changed since the time specified in the `OPTION_LQ_START_TIME` option. The binding information sent to satisfy this request is the catch-up information, and the period while it is being sent is the catch-up phase.

- o "clock skew"

The difference between the absolute time on a DHCPv6 server and the absolute time on the system where a requestor of an Active or Bulk Leasequery is executing is termed the "clock skew" for that Active or Bulk Leasequery connection. It is not absolutely constant but is likely to vary only slowly. While it is easy to think that this can be calculated precisely after one packet is received by a requestor from a DHCPv6 server, a more accurate value is derived from continuously examining the instantaneous value developed from each packet received from a DHCPv6 server and using it to make small adjustments to the existing value held in the requestor.

- o "Transaction ID"

An opaque value used to match responses with queries initiated by an Active Leasequery requestor.

3. Protocol Overview

The Active Leasequery mechanism is modeled on the existing DHCPv6 Bulk Leasequery [[RFC5460](#)]; most differences arise from the long term nature of the TCP [[RFC4614](#)] connection required for Active Leasequery. In addition, a DHCPv6 server which supports Active Leasequery MUST support Bulk Leasequery [[RFC5460](#)] as well.

An Active Leasequery requestor opens a TCP connection to a DHCPv6 Server, using the DHCPv6 port 547. Note that this implies that the Leasequery requestor has server IP address(es) available via configuration or some other means, and that it has unicast IP reachability to the DHCPv6 server. No relaying for Active Leasequery is specified.

After establishing a connection, the requestor sends an ACTIVELEASEQUERY message over the connection. In response, the server sends updates to the requestor using LEASEQUERY-REPLY and LEASEQUERY-DATA messages. This response procedure is identical to [[RFC5460](#)], except that in the case of Active Leasequery the server sends updates whenever some activity occurs to change the binding state - thus the need for long lived connection.

Active Leasequery has features which allow this external entity to lose its connection and then reconnect and receive the latest information concerning any IPv6 bindings changed while it was not connected.

These capabilities are designed to allow the Active Leasequery requestor to efficiently become current with respect to the lease state database after it has been restarted or the machine on which it is running has been reinitialized. It is easy to define a protocol which works when the requestor is always connected to the DHCPv6 server. Since that isn't sufficiently robust, much of the mechanism in this document is designed to deal efficiently with situations that occur when the Active Leasequery requestor becomes disconnected from the DHCPv6 server from which it is receiving updates and then becomes reconnected to that server.

Central to this approach, if the Active Leasequery requestor loses service, it is allowed to specify the time of its most recent update in a subsequent Active Leasequery request and the DHCPv6 server will

determine whether or not data was missed while the Active Leasequery requestor was not connected.

The DHCPv6 server processing the Active Leasequery request may limit the amount of data saved, and methods exist for the DHCPv6 server to inform the Active Leasequery requestor that more data was missed than could be saved. In this situation, the Active Leasequery requestor would issue a Bulk Leasequery [[RFC5460](#)] to recover information not available through an Active Leasequery.

DHCPv6 servers are not required to keep any data corresponding to data missed on an Active Leasequery connection, but will typically choose to keep data corresponding to some recent activity available for subsequent queries by a DHCPv6 Active Leasequery requestor whose connection was temporarily interrupted. In other words, DHCPv6 servers supporting catch-up are required to have some mechanism to keep/save historic information of bindings.

An Active Leasequery requestor would typically use Bulk Leasequery to initialize its database with all current data when that database contains no binding information. In addition, it would use Bulk Leasequery to recover missed information in the event that its connection with the DHCPv6 server was lost for a longer time than the DHCPv6 server would keep track of the specific changes to the IPv6 binding information.

The messages sent by the server in response to an Active Leasequery request SHOULD be identical to the messages sent by the server to a Bulk Leasequery request regarding the way the data is encoded into the Active Leasequery responses. In addition, the actions taken by the Active Leasequery requestor to interpret the responses to an Active Leasequery request SHOULD be identical to the way that the requestor interprets the responses to a Bulk Leasequery request. Thus, the handling of `OPTION_CLIENT_DATA` and additional options discussed in the Bulk Leasequery specification [[RFC5460](#)] are to be followed when implementing Active Leasequery.

4. Interaction Between Active Leasequery and Bulk Leasequery

Active Leasequery can be seen as an extension of the Bulk Leasequery protocol [[RFC5460](#)]. The format of packets returned to an Active Leasequery requestor are identical to that defined for the Bulk Leasequery protocol [[RFC5460](#)].

Applications which employ Active Leasequery to keep a database up to date with respect to the DHCPv6 server's lease state database will usually use an initial Bulk Leasequery to bring their database into equivalence with that of the DHCPv6 server, and then use Active

Leasequery to keep that database current with respect to the DHCPv6 server's lease state database.

There are several differences between the Active and Bulk Leasequery protocols. Active Leasequery defines a new message (ACTIVELEASEQUERY) to send Active Leasequery request to DHCPv6 server. An Active Leasequery connection sends all available updates to the requestor, based on OPTION_LQ_QUERY option (see [Section 6.2.1](#)).

An Active Leasequery connection does not ever "complete", though the DHCPv6 server may drop the connection for a variety of reasons associated with some sort of exception condition.

5. Extension to DHCPv6 Bulk Leasequery

This document extends to the capabilities of DHCPv6 Bulk Leasequery protocol [[RFC5460](#)] by defining new options (OPTION_LQ_BASE_TIME, OPTION_LQ_START_TIME and OPTION_LQ_END_TIME). DHCPv6 server sends OPTION_LQ_BASE_TIME option in Bulk Leasequery response if requestor ask for the same in Bulk Leasequery request. OPTION_LQ_START_TIME and OPTION_LQ_END_TIME can be used in Bulk Leasequery request made to DHCPv6 server. More details about these options are specified in [Section 6.3](#).

6. Message and Option Definitions

6.1. Message Framing for TCP

The use of TCP for the Active Leasequery protocol permits one or more DHCPv6 messages to be sent at a time. The receiver needs to be able to determine how large each message is. The same message framing technique used for DHCPv6 Bulk Leasequery [[RFC5460](#)] is used for Active Leasequery as well.

The intent in using the same format is that code which currently knows how to deal with a message returned from DHCPv6 Bulk Leasequery [[RFC5460](#)] will be able to deal with the message held inside of the TCP framing.

6.2. Messages

The LEASEQUERY-REPLY message is defined in [[RFC5007](#)]. The LEASEQUERY-DATA and LEASEQUERY-DONE messages are defined in [[RFC5460](#)].

In an Active Leasequery exchange, a single LEASEQUERY-REPLY message is used to indicate the success or failure of a query, and to carry

data that do not change in the context of a single query and answer, such as the Server-ID and Client-ID options. If a query is successful, only a single LEASEQUERY-REPLY message MUST appear. If the server is returning binding data, the LEASEQUERY-REPLY also contains the first client's binding data in an OPTION_CLIENT_DATA option. Additional binding data is returned using LEASEQUERY-DATA message as explained in DHCPv6 Bulk Leasequery [[RFC5460](#)]. In case of failure query, single LEASEQUERY-REPLY message is returned without any binding data.

6.2.1. ACTIVELEASEQUERY

The new message type (ACTIVELEASEQUERY) is designed for keeping the requestor up to date in real-time (or near real-time) with DHCPv6 bindings. It asks the server to return DHCPv6 bindings activity that occurs subsequent to the receipt of the Active Leasequery request.

An ACTIVELEASEQUERY request MUST contain a transaction-id, and that transaction-id MUST BE locally unique to the TCP connection to the DHCPv6 server.

When sending an Active Leasequery request, the requestor MAY include the OPTION_LQ_START_TIME option in the ACTIVELEASEQUERY request. In this case, DHCPv6 server returns all the bindings changed on or after the OPTION_LQ_START_TIME.

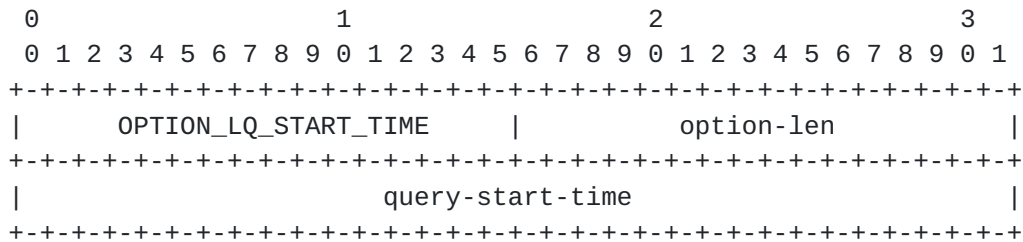
If the requestor is interested in receiving all binding updates from the DHCPv6 server, it MUST NOT include the OPTION_LQ_QUERY option in the ACTIVELEASEQUERY message. But if the requestor is only interested in specific binding updates, it MAY include an OPTION_LQ_QUERY option along with a query-types defined in [[RFC5007](#)] and [[RFC5460](#)].

Other DHCPv6 options used in the LEASEQUERY message (as specified in [[RFC5460](#)]) can also be used in the ACTIVELEASEQUERY request.

6.3. Options

New options (OPTION_LQ_BASE_TIME, OPTION_LQ_START_TIME and OPTION_LQ_END_TIME) are defined as an extension to DHCPv6 Bulk Leasequery [[RFC5460](#)]. The reply messages for Active Leasequery uses these options along with the options defined in [[RFC3315](#)], [[RFC5007](#)] and [[RFC5460](#)].

The code for this option is TBD.



option-code OPTION_LQ_START_TIME (TBD).
option-len 4.
query-start-time DHCPv6 Server Query Start Time.

6.3.3. OPTION_LQ_END_TIME

The OPTION_LQ_END_TIME option specifies a query end time to the DHCPv6 server. If specified, only bindings that have changed on or before the OPTION_LQ_END_TIME should be included in the response to the query. This option MAY be used in a Bulk Leasequery request. But it MUST NOT be used in an Active Leasequery request.

The requestor MUST determine the OPTION_LQ_END_TIME based on lease information it has received from the DHCPv6 server. This MUST be an absolute time in the context of the DHCPv6 server.

In the absence of information to the contrary, the requestor SHOULD assume that the time context of the DHCPv6 server is identical to the time context of the requestor (see [Section 8.4](#)).

This is an unsigned integer in network byte order.

The code for this option is TBD.


```

      0             1             2             3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          OPTION_LQ_END_TIME          |          option-len          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     query-end-time                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

option-code      OPTION_LQ_END_TIME (TBD).
option-len       4.
query-end-time   DHCPv6 Server Query End Time.

```

6.4. Connection and Transmission Parameters

Active Leasequery uses the same port configuration as DHCPv6 Bulk Leasequery [RFC5460]. It also uses the other transmission parameters (BULK_LQ_DATA_TIMEOUT and BULK_LQ_MAX_CONNS) as defined in [RFC5460].

This section presents a table of values used to control Active Leasequery behavior, including recommended defaults. Implementations MAY make these values configurable. However, configuring too-small timeout values may lead to harmful behavior both to this application as well as to other traffic in the network. As a result, timeout values smaller than the default values SHOULD NOT be used.

Parameter	Default	Description
ACTIVE_LQ_RCV_TIMEOUT	120 secs	Active Leasequery receive timeout
ACTIVE_LQ_SEND_TIMEOUT	120 secs	Active Leasequery send timeout
ACTIVE_LQ_IDLE_TIMEOUT	60 secs	Active Leasequery idle timeout

7. Information Communicated by Active Leasequery

While the information communicated by a DHCPv6 Bulk Leasequery [RFC5460] is taken directly from the DHCPv6 server's lease state database, the information communicated by an Active Leasequery is real-time information. As such, it is the information which is currently associated with a particular binding in the DHCPv6 server's lease state database.

This is of significance, because if the Active Leasequery requestor runs slowly or the requestor disconnects from the DHCPv6 server and

then reconnects with an `OPTION_LQ_START_TIME` option (signaling a catch-up operation), the information communicated to the Active Leasequery requestor is only the most current information from the DHCPv6 server's lease state database.

The requestor of an Active Leasequery **MUST NOT** assume that every lease state change is communicated across an Active Leasequery connection. Even if the Active Leasequery requestor remains connected, the DHCPv6 server is only required to transmit information about a binding that is current when the packet is created and handed off to the TCP stack to send to the requestor.

If the TCP connection blocks and the DHCPv6 server is waiting to send information down the connection, when the connection becomes available to be written the DHCPv6 server **MAY** create the packet to send at this time. The current state of the binding will be sent, and any transition in state or other information that occurred while the TCP connection was blocked will be lost.

Thus, the Active Leasequery protocol does not allow the requestor to build a complete history of every activity on every lease. An effective history of the important state changes for a lease can be created if the parameters of the DHCPv6 server are tuned to take into account the requirements of an Active Leasequery requestor. For instance, the period after the expiration or release of a binding could be configured long enough (say several minutes, well more than the receive timeout), so that an Active Leasequery requestor would be less likely to miss any changes in the binding.

8. Requestor Behavior

8.1. Connecting and General Processing

A Requestor attempts to establish a TCP connection to a DHCPv6 Server in order to initiate an Active Leasequery exchange. If the attempt fails, the Requestor **MAY** retry.

If an Active Leasequery is terminated prematurely by a `LEASEQUERY-DONE` with a DHCPv6 status code (carried in an `OPTION_STATUS_CODE` option) of `QueryTerminated` or by the failure of the connection over which it was being submitted, the requestor **MAY** retry the request after the creation of a new connection.

Messages from the DHCPv6 server come as multiple responses to a single `ACTIVELEASEQUERY` message. Thus, each `ACTIVELEASEQUERY` request **MUST** have an `xid` (transaction-id) unique on the connection on which it is sent, and all of the messages which come as a response to it contain the same `xid` as the request. It is the `xid` which allows the

data-streams of two or more different ACTIVELEASEQUERY requests to be de-multiplexed by the requestor.

A requestor MAY send an ACTIVELEASEQUERY request to a DHCPv6 server and immediately close the transmission side of its TCP connection, and then read the resulting response messages from the DHCPv6 server. This is not required, and the usual approach is to leave both sides of the TCP connection up until at least the conclusion of the Active Leasequery.

8.2. Forming an Active Leasequery

The Active Leasequery is designed to create a long lived connection between the requestor and the DHCPv6 server processing the active query. The DHCPv6 server will send binding information back across this connection with minimal delay after it learns of the binding information. It will learn about bindings either because it makes the bindings itself or because it has received information about a binding from another server.

To form the Active Leasequery, a DHCPv6 request is constructed with a message type of ACTIVELEASEQUERY. The DHCPv6 request MUST contain a transaction-id, and that transaction-id MUST BE locally unique to the TCP connection to the DHCPv6 server.

An important capability of the Active Leasequery is the ability of the requestor to specify that some recent data be sent immediately to the requestor in parallel with the transmission of the ongoing binding information in more or less real time. This capability is used in order to allow an Active Leasequery requestor to recover missed information in the event that it temporarily loses connectivity with the DHCPv6 server processing a previous Active Leasequery.

Note that until all of the recent data (catch-up data) has been received, the requestor MUST NOT keep track of the base-time (OPTION_LQ_BASE_TIME) received in Leasequery reply messages to use later in a subsequent Active Leasequery request.

This capability is enabled by the transmission of an OPTION_LQ_BASE_TIME option with each Leasequery reply sent as the result of a previous Active Leasequery. The requestor will typically keep track of the highest base-time received from a particular DHCPv6 server over an Active Leasequery connection, and in the event that the requestor finds it necessary (for whatever reason) to reestablish an Active Leasequery connection to that DHCPv6 server, the requestor will place this highest base-time value into an OPTION_LQ_START_TIME option in the new Active Leasequery request.

If the requestor doesn't wish to request an update of information missed when it was not connected to the DHCPv6 server, then it does not include the `OPTION_LQ_START_TIME` option in the Active Leasequery request.

If the TCP connection becomes blocked or stops being writable while the requestor is sending its query, the requestor **SHOULD** be prepared to terminate the connection after `BULK_LQ_DATA_TIMEOUT`. We make this recommendation to allow requesters to control the period of time they are willing to wait before abandoning a connection, independent of notifications from the TCP implementations they may be using.

8.3. Processing Active Replies

The Requestor attempts to read a DHCPv6 `LEASEQUERY-REPLY` message from the TCP connection. If the stream of replies becomes blocked, the Requestor **SHOULD** be prepared to terminate the connection after `ACTIVE_LQ_RCV_TIMEOUT`, and **MAY** begin retry processing if configured to do so.

The requestor examines the `LEASEQUERY-REPLY` message, and determines how to proceed. Message validation rules are specified in DHCPv6 Leasequery [[RFC5007](#)] and DHCPv6 Bulk Leasequery [[RFC5460](#)]. If the reply contains an DHCPv6 status code (carried in an `OPTION_STATUS_CODE` option), the requestor follows the recommendations in [[RFC5007](#)].

Note that an Active Leasequery request specifically requests the DHCPv6 server to create a long-lived connection which may not have data transferring continuously during its lifetime. Therefore the DHCPv6 server will send a `LEASEQUERY-DATA` message without binding data (`OPTION_CLIENT_DATA`) every `ACTIVE_LQ_IDLE_TIMEOUT` seconds (default 60) in order for the requestor to know that the connection remains alive. This approach is followed only when connection is idle (i.e. server has no binding data to send). During normal binding data exchange, receiving of `LEASEQUERY-DATA` message by requestor itself signifies that connection is active. Note that the default for `ACTIVE_LQ_RCV_TIMEOUT` is 120 seconds, twice the value of the `ACTIVE_LQ_IDLE_TIMEOUT`'s default of 60 seconds which drives the DHCPv6 server to send messages. Thus `ACTIVE_LQ_RCV_TIMEOUT` controls how sensitive the requestor is to be to delays by the DHCPv6 server in sending updates or `LEASEQUERY-DATA` messages.

A single Active Leasequery can and usually will result in a large number of replies. The Requestor **MUST** be prepared to receive more than one reply with transaction-ids matching a single `ACTIVELEASEQUERY` message from a single DHCPv6 server.

An Active Leasequery has two regimes -- during the catch-up phase, if any, and after any catch-up phase. During the catch-up phase (if one exists), the data returned in the `OPTION_LQ_BASE_TIME` option in a `LEASEQUERY-REPLY` or `LEASEQUERY-DATA` message may appear to be ordered, but the most recent change in the lease state data being returned is not related to the `OPTION_LQ_BASE_TIME` option value in the messages. Another way to say this is that the ordering of the updates sent by the DHCPv6 server during the catch-up phase is independent of the ordering in the changes in the lease state data. The `OPTION_LQ_BASE_TIME` option from messages during this phase **MUST NOT** be saved and used in a subsequent `ACTIVELEASEQUERY` message's `OPTION_LQ_START_TIME` option as it does not represent the extent of progress of the catch-up activity.

After the catch-up phase, or during the entire series of messages received as the response to an Active Leasequery request with no `OPTION_LQ_START_TIME` (and therefore no catch-up phase), the `OPTION_LQ_BASE_TIME` option of the most recent message **SHOULD** be saved as a record of the most recent time that data was received. This base-time (in the context of the DHCPv6 server) can be used in a subsequent Active Leasequery message's `OPTION_LQ_START_TIME` after a loss of the Active Leasequery connection.

The `LEASEQUERY-DONE` message **MAY** unilaterally terminate a successful Active Leasequery request which is currently in progress in the event that the DHCPv6 server determines that it cannot continue processing a `ACTIVELEASEQUERY` request. For example, when a server is requested to shut down it **SHOULD** send a `LEASEQUERY-DONE` message with a DHCPv6 status code of `QueryTerminated` and include `OPTION_LQ_BASE_TIME` option in the message. This **SHOULD** be the last message on that connection, and once the message has been transmitted, the server should close the connection.

After receiving `LEASEQUERY-DONE` with a `QueryTerminated` status from a server, the Requestor **MAY** close the TCP connection to that server.

8.3.1. Processing Replies from a Request Containing a `OPTION_LQ_START_TIME`

If the Active Leasequery was requested with an `OPTION_LQ_START_TIME` option, the DHCPv6 server will attempt to send information about all bindings that changed since the time specified in the `OPTION_LQ_START_TIME`. This is the catch-up phase of the Active Leasequery processing. The DHCPv6 server **MAY** also begin immediate updates over the same connection of real-time binding information changes. Thus, the catch-up phase may run in parallel with the normal updates generated by the Active Leasequery request.

A DHCPv6 server MAY keep only a limited amount of time ordered information available to respond to an Active Leasequery request containing an `OPTION_LQ_START_TIME` option. Thus, it is possible that the time specified in the `OPTION_LQ_START_TIME` option represents a time not covered by the time ordered information kept by the DHCPv6 server. If this should occur, and there is not enough data saved in the DHCPv6 server to satisfy the request specified by the `OPTION_LQ_START_TIME` option, the DHCPv6 server will reply immediately with a `LEASEQUERY-REPLY` message with a DHCPv6 status code of `DataMissing` with a base-time option equal to the server's current time. This will signal the end of the catch-up phase, and the only updates that will subsequently be received on this connection are the real-time updates from the Active Leasequery request.

If there is enough data saved to satisfy the request, then `LEASEQUERY-REPLY` (with `OPTION_STATUS_CODE` of `Success` or reply without `OPTION_STATUS_CODE` option) and `LEASEQUERY-DATA` messages will begin arrive from the DHCPv6 server. Some of these messages will be related to the `OPTION_LQ_START_TIME` request and be part of the catch-up phase. Some of these messages will be real-time updates of binding changes taking place in the DHCPv6 server. In general, there is no way to determine the source of each message.

Until the catch-up phase is complete, the latest base-time value received from a DHCPv6 server processing an Active Leasequery request cannot be reset from the incoming messages because to do so would compromise the ability to recover lost information if the Active Leasequery were to terminate prior to the completion of the catch-up phase.

The requestor will know that the catch-up phase is complete when the DHCPv6 server will transmit a `LEASEQUERY-DATA` message with the DHCPv6 status code of `CatchUpComplete`. Once this message is transmitted, all additional `LEASEQUERY-DATA` messages will relate to real-time ("new") binding changes in the DHCPv6 server.

As discussed in [Section 8.3](#), the requestor SHOULD keep track of the latest base-time option value received over a particular connection, to be used in a subsequent Active Leasequery request -- but only if the catch-up phase is complete. Prior to the completion of the catch-up phase, if the connection should go away or if the requestor receives a `LEASEQUERY-DONE` message, then when it reconnects it MUST use the base-time value from the previous connection and not any base-time value received from the recently closed connection.

In the event that there was enough data available to the DHCPv6 server to begin to satisfy the request implied by the `OPTION_LQ_START_TIME` option, but during the processing of that data

the server found that it was unable to continue (perhaps there was barely enough, the connection is very slow, and the aging algorithm causes the saved data to become unavailable) the DHCPv6 server will terminate the catch-up phase of processing immediately by sending a LEASEQUERY-DATA message with a DHCPv6 status code of DataMissing and with a base-time option of the current time.

The requestor MUST NOT assume that every individual state change of every binding during the period from the time specified in the OPTION_LQ_START_TIME and the present is replicated in an Active Leasequery reply message. The requestor MAY assume that at least one Active Leasequery reply message will exist for every binding which had one or more changes of state during the period specified by the OPTION_LQ_START_TIME and the current time. The last message for each binding will contain the state at the current time, and there may be one or more messages concerning a single binding during the catch-up phase of processing.

If a binding changed state multiple times during the time that the requestor was not connected (that is, during the time from the OPTION_LQ_START_TIME and the present), then only the current binding information will be sent during the catch-up phase. However, the requestor MUST NOT assume that every intermediate state change that occurred during the period from the OPTION_LQ_START_TIME to the present will be represented by an individual Leasequery message.

If the LEASEQUERY-REPLY or LEASEQUERY-DATA message containing a DHCPv6 status code of DataMissing is received and the requestor is interested in keeping its database up to date with respect to the current state of bindings in the DHCPv6 server, then the requestor SHOULD issue a Bulk Leasequery request to recover the information missing from its database. This Bulk Leasequery request should include a OPTION_LQ_START_TIME option with the same value as the OPTION_LQ_START_TIME option previously included in the ACTIVELEASEQUERY responses from the DHCPv6 server, and an OPTION_LQ_END_TIME option equal to the OPTION_LQ_BASE_TIME option returned by the DHCPv6 server in the LEASEQUERY-REPLY or LEASEQUERY-DATA message with the DHCPv6 status code of DataMissing.

In the event that the requestor receives a LEASEQUERY-REPLY or LEASEQUERY-DATA message with a DHCPv6 status code of DataMissing, it is a reasonable assumption that it is interested in keeping its database up to date with respect to the DHCPv6 server's internal binding database or it would not have included the OPTION_LQ_START_TIME in the ACTIVELEASEQUERY message.

Typically, the requestor would have one connection open to a DHCPv6 server for an Active Leasequery request and possibly one additional

connection open for a Bulk Leasequery request to the same DHCPv6 server to fill in the data that might have been missed prior to the initiation of the Active Leasequery. The Bulk Leasequery connection would typically run to completion and be closed, leaving one Active Leasequery connection open to a single DHCPv6 server. Alternatively, both requests could be issued over a single connection.

8.4. Processing Time Values in Leasequery messages

Active or Bulk Leasequery requests may be made to a DHCPv6 server whose absolute time may not be synchronized with the local time of the requestor. Thus, there are at least two time contexts in even the simplest Active or Bulk Leasequery response.

If the requestor of an Active or Bulk Leasequery is saving the data returned in some form, it has a requirement to store a variety of time values, and some of these will be time in the context of the requestor and some will be time in the context of the DHCPv6 server.

When receiving an Active or Bulk Leasequery reply message from the DHCPv6 server, the message will contain an `OPTION_LQ_BASE_TIME` option. The time contained in this `OPTION_LQ_BASE_TIME` option is in the context of the DHCPv6 server. As such, it is an ideal time to save and use as input to an Active or Bulk Leasequery message in the `OPTION_LQ_START_TIME` or `OPTION_LQ_END_TIME` option, should the requestor need to ever issue an Active or Bulk Leasequery message using these option as part of a later query, since these option requires a time in the context of the DHCPv6 server.

In addition to saving the `OPTION_LQ_BASE_TIME` for possible future use in `OPTION_LQ_START_TIME` or `OPTION_LQ_END_TIME` option, the `OPTION_LQ_BASE_TIME` is used as part of the conversion of the other times in the Leasequery message to values which are meaningful in the context of the requestor.

In systems whose clocks are synchronized, perhaps using NTP, the clock skew will usually be zero, which is not only acceptable, but desired.

8.5. Examples

These examples illustrate what a series of queries and responses might look like. These are only examples -- there are no requirement that these sequence must be followed.

8.5.1. Query Failure

This example illustrates the message flows in case DHCPv6 server identifies that it cannot accept and/or process Active Leasequery request from the requestor. This could be because of various reasons (i.e. UnknownQueryType, MalformedQuery, NotConfigured, NotAllowed).

```

Client                               Server
-----
ACTIVELEASEQUERY xid 1  ----->
                           <----- LEASEQUERY-REPLY xid 1 (w/error)

```

8.5.2. Data Missing on Server

This example illustrates the message flows in case DHCPv6 server identifies that it does not have enough data saved to satisfy the request specified by the `OPTION_LQ_START_TIME` option.

In this case the DHCPv6 server will reply immediately with a `LEASEQUERY-REPLY` message with a DHCPv6 status code of `DataMissing` with a base-time option equal to the server's current time. This will signal the end of the catch-up phase, and the only updates that will subsequently be received on this connection are the real-time updates from the Active Leasequery request.

```

Client                               Server
-----
ACTIVELEASEQUERY xid 2  ----->
                           <----- LEASEQUERY-REPLY xid 2 (w/error)
                           <----- LEASEQUERY-DATA xid 2
                           <----- LEASEQUERY-DATA xid 2
                           <----- LEASEQUERY-DATA xid 2

```

8.5.3. Successful Query

This example illustrates the message flows in case of successful query processing by DHCPv6 server.

In this case the DHCPv6 server will reply immediately with a `LEASEQUERY-REPLY` message (with `OPTION_STATUS_CODE` of `Success` or reply without `OPTION_STATUS_CODE` option), followed by binding data in `LEASEQUERY-DATA` messages. In case, DHCPv6 server wants to abort in-process request and terminate the connection due to some reason, it sends `LEASEQUERY-DONE` with error code present in `OPTION_STATUS_CODE` option.


```
Client                               Server
-----                               -
ACTIVELEASEQUERY xid 3  <----->
                           <----- LEASEQUERY-REPLY xid 3
                           <----- LEASEQUERY-DATA xid 3
                           <----- LEASEQUERY-DATA xid 3
                           <----- LEASEQUERY-DATA xid 3
                           <----- LEASEQUERY-DATA xid 3
                           <----- LEASEQUERY-DONE xid 3 (w/error)
```

8.6. Closing Connections

The Requestor or DHCPv6 Leasequery server MAY close its end of the TCP connection at any time. The Requestor MAY choose to retain the connection if it intends to issue additional queries. Note that this requestor behavior does not guarantee that the connection will be available for additional queries: the server might decide to close the connection based on its own configuration.

9. Server Behavior

A DHCPv6 server which supports Active Leasequery MUST support DHCPv6 Bulk Leasequery [[RFC5460](#)] and as extended herein.

9.1. Accepting Connections

Servers that implement DHCPv6 Active Leasequery listen for incoming TCP connections. Approach used in accepting (or rejecting) the requestor connection is same as specified in DHCPv6 Bulk Leasequery [[RFC5460](#)].

9.2. Replying to an Active Leasequery

The DHCPv6 Leasequery [[RFC5007](#)] specification describes the initial construction of LEASEQUERY-REPLY messages. Use of the LEASEQUERY-REPLY and LEASEQUERY-DATA messages to carry multiple bindings is described in DHCPv6 Bulk Leasequery [[RFC5460](#)]. Message transmission and framing for TCP is described in [Section 6.1](#).

If the connection becomes blocked while the server is attempting to send reply messages, the server SHOULD be prepared to terminate the TCP connection after ACTIVE_LQ_SEND_TIMEOUT. This timeout governs how much congestion the DHCPv6 server is prepared to tolerate over any Active Leasequery connection. The default is two minutes, which means that if more than two minutes goes by without the requestor reading enough information to unblock the TCP connection, the DHCPv6 server will drop the TCP connection.

If the DHCPv6 server encounters an error during initial processing of the ACTIVELEASEQUERY message, it SHOULD send a LEASEQUERY-REPLY message containing an error code of some kind in a DHCPv6 status code option. It SHOULD close the connection after this error is signaled.

If the DHCPv6 server encounters an error during later processing of the ACTIVELEASEQUERY message, it SHOULD send a LEASEQUERY-DONE containing an error code of some kind in a DHCPv6 status code option. It SHOULD close the connection after this error is signaled.

If the server finds any bindings satisfying a query, it sends each binding's data in a reply message. The first reply message is a LEASEQUERY-REPLY. The binding data is carried in an OPTION_CLIENT_DATA option, as specified in [\[RFC5007\]](#). The server returns subsequent bindings in LEASEQUERY-DATA messages, which can avoid redundant data (such as the requestor's Client-ID).

Every reply to an Active Leasequery request MUST contain the information specified in replies to a DHCPv6 Bulk Leasequery request [\[RFC5460\]](#).

If an Active Leasequery or Bulk Leasequery request contains OPTION_LQ_BASE_TIME option code present in OPTION_ORO, the DHCPv6 server MUST include OPTION_LQ_BASE_TIME option in every reply for this request. The value for base-time option is current absolute time in the DHCPv6 server's context.

If an Active Leasequery request contains an OPTION_LQ_START_TIME option, it indicates that the requestor would like the DHCPv6 server to send it not only messages that correspond to DHCPv6 binding activity that occurs subsequent to the receipt of the Active Leasequery request, but also messages that correspond to DHCPv6 binding activity that occurred prior to the Active Leasequery request.

If OPTION_LQ_END_TIME option appears in an Active Leasequery request, the DHCPv6 server should send a LEASEQUERY-REPLY message with a DHCPv6 status code of MalformedQuery and terminate the connection.

In order to implement a meaningful response to this query, the DHCPv6 server MAY keep track of the binding activity and associate changes with particular base-time values from the messages. Then, when requested to do so by an Active Leasequery request containing a OPTION_LQ_START_TIME option, the DHCPv6 server can respond with replies for all binding activity occurring on that OPTION_LQ_START_TIME or later times.

These replies based on the `OPTION_LQ_START_TIME` MAY be interleaved with the messages generated due to current binding activity.

Once the transmission of the DHCPv6 Leasequery messages associated with the `OPTION_LQ_START_TIME` option are complete, a `LEASEQUERY-DATA` message MUST be sent with a DHCPv6 status code value of `CatchUpComplete`.

The DHCPv6 server SHOULD, but is not required to, keep track of a limited amount of previous binding activity. The DHCPv6 server MAY choose to only do this in the event that it has received at least one Active Leasequery request in the past, as to do so will almost certainly entail some utilization of resources which would be wasted if there are no Active Leasequery requestors for this DHCPv6 server. The DHCPv6 server SHOULD make the amount of previous binding activity it retains configurable. There is no requirement on the DHCPv6 server to retain this information over a server restart (or even to retain such information at all).

Unless there is an error or some requirement to cease processing a Active Leasequery request yielding a `LEASEQUERY-DONE` message, such as a server shutdown, there will be no `LEASEQUERY-DONE` message at the conclusion of the Active Leasequery processing because that processing will not conclude but will continue until either the requestor or the server drops the connection.

9.3. Multiple or Parallel Queries

Requesters may want to use an existing connection if they need to make multiple queries. Servers MAY support reading and processing multiple queries from a single connection. A server MUST NOT read more query messages from a connection than it is prepared to process simultaneously.

Typically, a requestor of an Active Leasequery would not need to send a second Active Leasequery while the first is still active. However, sending an Active Leasequery and a Bulk Leasequery over the same connection would be possible and reasonable. But it is RECOMMENDED to use different connection in case of parallel Active and Bulk Leasequeries.

This MAY be a feature that is administratively controlled. Servers that are able to process queries in parallel SHOULD offer configuration that limits the number of simultaneous queries permitted from any one requestor, in order to control resource use if there are multiple requesters seeking service.

9.4. Closing Connections

The server MUST close its end of the TCP connection if it encounters an error sending data on the connection. The server MUST close its end of the TCP connection if it finds that it has to abort an in-process request. A server aborting an in-process request SHOULD attempt to signal that to its requestors by using the QueryTerminated status code in the DHCPv6 status code option in a LEASEQUERY-DONE message. If the server detects that the requestor end has been closed, the server MUST close its end of the connection after it has finished processing any outstanding requests from the requestor.

The server SHOULD be prepared to limit the number of connections it maintains, and SHOULD be prepared to close idle connections to enforce the limit.

10. Security Considerations

The "Security Considerations" section of [[RFC3315](#)] details the general threats to DHCPv6. The DHCPv6 Leasequery specification [[RFC5007](#)] describes recommendations for the Leasequery protocol, especially with regard to relayed Leasequery messages, mitigation of packet-flooding denial-of-service (DoS) attacks, restriction to trusted requestors, and use of IPsec [[RFC4301](#)].

The use of TCP introduces some additional concerns. Attacks that attempt to exhaust the DHCPv6 server's available TCP connection resources, such as SYN flooding attacks, can compromise the ability of legitimate requestors to receive service. Malicious requestors who succeed in establishing connections, but who then send invalid queries, partial queries, or no queries at all also can exhaust a server's pool of available connections. We recommend that servers offer configuration to limit the sources of incoming connections, that they limit the number of accepted connections and the number of in-process queries from any one connection, and that they limit the period of time during which an idle connection will be left open.

There are two specific issues regarding Active Leasequery security that deserve explicit mention. The first is preventing information that Active Leasequery can provide from reaching requestors who are not authorized to receive such information. The second is ensuring that authorized requestors of the Active Leasequery capability receive accurate information from the Server (and that this information is not disrupted in transit).

To prevent information leakage to unauthorized requestors, Servers SHOULD restrict Active Leasequery connections and ACTIVELEASEQUERY messages to certain requestors, either through explicit configuration

of the Server itself or by employing external network elements to provide such restrictions. In particular, the typical DHCPv6 client SHOULD NOT be allowed to receive a response to an Active Leasequery request, and some technique MUST exist to allow prevention of such access in any environment where Active Leasequery is deployed.

Connections not from permitted requestors SHOULD be closed immediately, to avoid server connection resource exhaustion or alternatively, simply not be allowed to reach the server at all. Servers SHOULD have the capability to restrict certain requestors to certain query types. Servers MAY reply to queries that are not permitted with the LEASEQUERY-DONE message with a status-code option status of NotAllowed, or MAY simply close the connection.

To prevent interception, disruption and malicious corruption of Active Leasequery data flows between the server and authorized requestors these data flows SHOULD transit only secured networks. These data flows are typically infrastructure oriented, and there is usually no reason to have them flowing over networks where such attacks are likely. In the rare cases where these data flows might need to be sent through unsecured networks, they MUST be sent over connections secured through means external to the DHCPv4/DHCPv6 server and its requestor(s) (e.g., through VPN's).

Authentication for DHCP Messages [[RFC3315](#)] MUST NOT be used to attempt to secure transmission of the messages described in this document.

11. IANA Considerations

IANA is requested to assign new DHCPv6 Option Codes in the registry maintained in <http://www.iana.org/assignments/dhcpv6-parameters>:

OPTION_LQ_BASE_TIME

OPTION_LQ_START_TIME

OPTION_LQ_END_TIME

IANA is requested to assign new values in the registry of DHCPv6 Status Codes maintained in <http://www.iana.org/assignments/dhcpv6-parameters>:

DataMissing

CatchUpComplete

IANA is requested to assign value for the following new DHCPv6 Message type in the registry maintained in <http://www.iana.org/assignments/dhcpv6-parameters>:

ACTIVELEASEQUERY

12. Acknowledgements

Some of the concept and content, present in this document, are based on DHCPv4 Active Leasequery which was originally proposed by Kim Kinnear, Bernie Volz, Mark Stapp and Neil Russell.

13. Modification History

14. References

14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3315](#), July 2003.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", [RFC 3633](#), December 2003.
- [RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng, "DHCPv6 Leasequery", [RFC 5007](#), September 2007.
- [RFC5460] Stapp, M., "DHCPv6 Bulk Leasequery", [RFC 5460](#), February 2009.

14.2. Informative References

- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", [RFC 4301](#), December 2005.
- [RFC4614] Duke, M., Braden, R., Eddy, W., and E. Blanton, "A Roadmap for Transmission Control Protocol (TCP) Specification Documents", [RFC 4614](#), September 2006.

Authors' Addresses

Dushyant Raghuvanshi
Cisco Systems, Inc.
Cessna Business Park,
Varthur Hobli, Outer Ring Road,
Bangalore, Karnataka 560037
India

Phone: +91 (080) 4365-7476
Email: draghuva@cisco.com

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

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

Deepak Kukrety
Cisco Systems, Inc.
Cessna Business Park,
Varthur Hobli, Outer Ring Road,
Bangalore, Karnataka 560037
India

Phone: +91 (080) 4365-7474
Email: dkukrety@cisco.com

