                        DHCPv6 Active Leasequery

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 updates DHCPv6 Bulk Leasequery (RFC 5460) by
   adding new options.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7653.

Copyright Notice

Table of Contents

1. Introduction

   The DHCPv6 protocol [RFC3315] specifies a mechanism for the
   assignment of IPv6 address and configuration information to IPv6
   nodes. IPv6 Prefix Delegation for DHCPv6 [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
   entities need to keep up with the current binding activity of the
   DHCPv6 server. Keeping up with this 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 that is directly involved
   in the DHCPv6 client-server transactions (e.g., a relay agent), or it
   could be an external process that 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.

   This document updates DHCPv6 Bulk Leasequery [RFC5460] by adding new
   options, as described in Section 6.2.1. For DHCPv6 servers
   supporting Bulk Leasequery and not Active Leasequery, Section 9.2
   specifies the mechanism to reject incoming Active Leasequery
   requests.

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 unsigned quantity containing the number of seconds since
      midnight (UTC), January 1, 2000, modulo 2^32.

o Active Leasequery

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

o Bulk Leasequery

   Requesting and receiving information about all or some of the
   existing DHCPv6 binding information in an efficient manner, as
   defined by [RFC5460].

o blocked TCP connection

   A TCP connection is considered blocked if the underlying TCP
   transport will not accept new messages to be sent without blocking
   the thread that is attempting to send the message.

o binding change/update

   Any change in the DHCPv6 binding state. This also includes
   expiration or deletion of the binding.

o catch-up information

   If a DHCPv6 Active Leasequery requestor sends an
   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.

o catch-up phase

   The period while catch-up information 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 message is
   received by a requestor from a DHCPv6 server, a more accurate
   value is derived from continuously examining the instantaneous
   value developed from each message received from a DHCPv6 server

and using it to make small adjustments to the existing value held
in the requestor.

o DHCPv6 binding state

   Data stored on the DHCPv6 server related to binding.

o requestor

   The node that sends LEASEQUERY messages to one or more servers to
   retrieve information on the bindings for a client.

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 [RFC7414] connection required for Active
   Leasequery. A DHCPv6 server that 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 similar to the
   procedure specified in [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 a long-lived
   connection. Additionally, the Active Leasequery server SHOULD
   provide a mechanism to control which data is allowed to be included
   in the OPTION_CLIENT_DATA messages sent to the requestor. See
   Section 9.3.

   Active Leasequery has features that 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 features 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 that
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 reconnects
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 data was missed (i.e.,
not all data could be saved). In this situation, the Active
Leasequery requestor should 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, with the exception that
a server responding to an Active Leasequery request SHOULD be able to
be configured to prevent specific data items from being included in
the OPTION_CLIENT_DATA option even if they were requested by
inclusion in the OPTION_ORO option.

4. Interaction between Active Leasequery and Bulk Leasequery

   Active Leasequery is an extension of the Bulk Leasequery protocol
   [RFC5460]. The format of messages returned to an Active Leasequery
   requestor is identical to that defined for the Bulk Leasequery
   protocol [RFC5460].

   Applications that employ Active Leasequery to keep a database up to
   date with respect to the DHCPv6 server's lease state database should
   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 requests to the DHCPv6
   server. An Active Leasequery connection sends all available updates
   to the requestor, based on the OPTION_LQ_QUERY option (see
   Section 6.2.1).

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

5. Extension to DHCPv6 Bulk Leasequery

   This document extends the capabilities of the DHCPv6 Bulk Leasequery
   protocol [RFC5460] by defining new options (OPTION_LQ_BASE_TIME,
   OPTION_LQ_START_TIME, and OPTION_LQ_END_TIME). The DHCPv6 server
   sends the OPTION_LQ_BASE_TIME option in a Bulk Leasequery response if
   the requestor asked for the same in the Bulk Leasequery request.
   OPTION_LQ_START_TIME and OPTION_LQ_END_TIME can be used in a Bulk
   Leasequery request made to the 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 in response to a single Active Leasequery
   request. 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 that 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.

   When using Transport Layer Security (TLS), once TLS negotiation
   completes, the connection will be encrypted and is now protected from
   eavesdropping, and normal Active Leasequery messages are sent and
   received using the TLS application data protocol services (see
   Section 10 of [RFC5246]).

6.2. Messages

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 binding 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 on the TCP connection on which
   it is sent to the DHCPv6 server.

   When sending an ACTIVELEASEQUERY request, the requestor MAY include
   the OPTION_LQ_START_TIME option in the ACTIVELEASEQUERY request. In
   this case, the 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 message.

6.2.2. STARTTLS

   The new message type (STARTTLS) is designed for establishment of a
   TLS connection between a requestor and a DHCPv6 server. The STARTTLS
   message SHOULD be sent without any options. Any options received in
   a STARTTLS message SHOULD be ignored.

   More details about this message are specified in Section 8.2.

6.2.3. Response 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, the DHCPv6 server MUST respond to it with exactly one
   LEASEQUERY-REPLY message. 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 a LEASEQUERY-DATA message as explained in DHCPv6 Bulk
   Leasequery [RFC5460]. In case of a query failure, a single
   LEASEQUERY-REPLY message is returned without any binding data.

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 use
   these options along with the options defined in [RFC3315], [RFC5007],
   and [RFC5460].

6.3.1. OPTION_LQ_BASE_TIME

   The OPTION_LQ_BASE_TIME option is the current time the message was
   created to be sent by the DHCPv6 server to the requestor of the
   Active or Bulk Leasequery if the requestor asked for the same in an
   Active or Bulk Leasequery request. This MUST be an absolute time
   (i.e., seconds since midnight January 1, 2000 UTC). All of the other
   time-based options in the reply message are relative to this time,
   including OPTION_CLT_TIME [RFC5007]. This time is in the context of
   the DHCPv6 server that placed this option in a message.

   This is an unsigned integer in network byte order.

   The code for this option is 100.

```
     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_BASE_TIME      |           option-len          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                            base-time                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
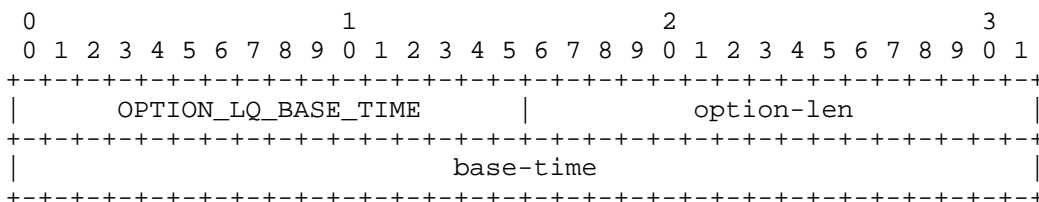
      option-code   OPTION_LQ_BASE_TIME (100)
      option-len   4
      base-time   DHCPv6 Server Base Time

6.3.2. OPTION_LQ_START_TIME

   The OPTION_LQ_START_TIME option specifies a query start time to the
   DHCPv6 server. If specified, only bindings that have changed on or
   after the OPTION_LQ_START_TIME should be included in the response to
   the query. This option MAY be used in Active or Bulk Leasequery
   requests made to a DHCPv6 server.

   The requestor MUST determine the OPTION_LQ_START_TIME using lease
   information it has received from the DHCPv6 server. This MUST be an
   absolute time in the DHCPv6 server's context (see Section 8.5).

   Typically (though this is not a requirement), the
   OPTION_LQ_START_TIME option will contain the value most recently
   received in an OPTION_LQ_BASE_TIME option by the requestor, as this
   will indicate the last successful communication with the DHCPv6
   server.

   This is an unsigned integer in network byte order.

   The code for this option is 101.

```
     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_START_TIME     |           option-len          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         query-start-time                      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

      option-code   OPTION_LQ_START_TIME (101)
      option-len   4
      query-start-time   DHCPv6 Server Query Start Time
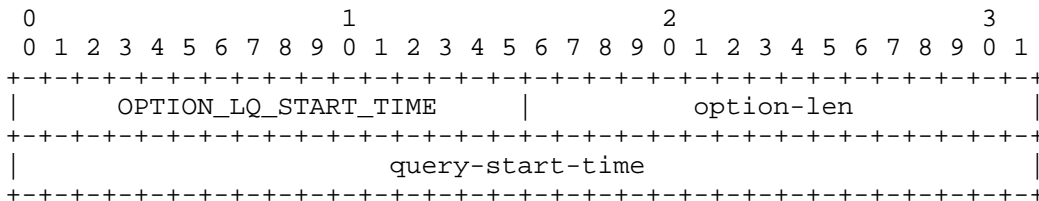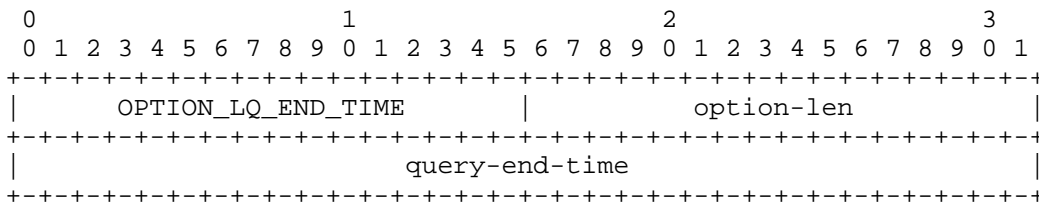
6.3.3. OPTION_LQ_END_TIME

   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 8.5).

   This is an unsigned integer in network byte order.

   The code for this option is 102.

```
    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 (102)
      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
   and 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 that 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 message 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 message 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. 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. Retries should not be more frequent
    than one every ACTIVE_LQ_IDLE_TIMEOUT. See Section 6.4.

    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. Retries should not be more
    frequent than one every ACTIVE_LQ_IDLE_TIMEOUT. See Section 6.4.

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

8.2. Initiating a Connection

    A requestor SHOULD be able to operate in either insecure or secure
    mode. This MAY be a feature that is administratively controlled.

    When operating in insecure mode, the requestor SHOULD proceed to send
    an ACTIVELEASEQUERY message after the establishment of a TCP
    connection.

    When operating in secure mode, the requestor MUST attempt to
    negotiate a TLS [RFC5246] connection over the TCP connection. If
    this negotiation fails, the requestor MUST close the TCP connection.
    The recommendations in [RFC7525] SHOULD be followed when negotiating
    this connection.

    A requestor requests the establishment of a TLS connection by sending
    the STARTTLS message to the DHCPv6 server as the first message over
    the TCP connection. This message indicates to the DHCPv6 server that
    a TLS connection over this TCP connection is desired. There are four
    possibilities after the requestor sends the STARTTLS message to the
    DHCPv6 server:

    1. No response from the DHCPv6 server.

   2. The DHCPv6 server closes the TCP connection after it receives the
       STARTTLS message.

   3. The DHCPv6 server responds with a REPLY [RFC3315] message with a
       DHCPv6 status code of TLSConnectionRefused.

   4. The DHCPv6 server responds with a REPLY [RFC3315] message without
       a DHCPv6 status code, indicating success.

   In any of the first three possibilities, the DHCPv6 server can be
   assumed to not support TLS. In this case, the requestor MUST close
   the TCP connection.

   In the final possibility, where the DHCPv6 server has responded with
   a REPLY message without a DHCPv6 status code in response to the
   requestor's STARTTLS message, the requestor SHOULD initiate the
   exchange of the messages involved in a TLS handshake [RFC5246].
   During the TLS handshake, the requestor MUST validate the DHCPv6
   server's digital certificate.

   If the handshake exchange yields a functioning TLS connection, then
   the requestor SHOULD transmit an ACTIVELEASEQUERY request over that
   TLS connection and use that TLS connection for all further
   interactions in which it engages with the DHCPv6 server over this TCP
   connection.

   If the handshake exchange does not yield a functioning TLS
   connection, then the requestor MUST close the TCP connection.

8.3. Forming an Active Leasequery

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

   An important capability of 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.

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 SHOULD 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
SHOULD place this highest base-time value into an
OPTION_LQ_START_TIME option in the new Active Leasequery request.

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.

If the requestor doesn't wish to request an update of information
missed when it was not connected to the DHCPv6 server, then it SHOULD
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 terminate
the connection after BULK_LQ_DATA_TIMEOUT. We make this
recommendation to allow requestors 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.

8.4. 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 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 a DHCPv6 status code (carried in an OPTION_STATUS_CODE
option), the requestor should follow the recommendations in
[RFC5007].

Note that the connection resulting from accepting an Active
Leasequery request may be long-lived and may not have data
transferring continuously during its lifetime. Therefore, the DHCPv6
server SHOULD 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 a normal exchange of
binding data, receiving a LEASEQUERY-DATA message 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 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. If the Active Leasequery was
requested with an OPTION_LQ_START_TIME option, the Active Leasequery
starts out in the catch-up phase. See Section 8.4.1 for information
on processing during the catch-up phase, as well as how to determine
when the catch-up phase is complete.

The updates sent by the DHCPv6 server during the catch-up phase are
not in the order that the lease state data was updated. Therefore,
the OPTION_LQ_BASE_TIME option from messages during this phase MUST
NOT be saved and used to compute the subsequent ACTIVELEASEQUERY
message's OPTION_LQ_START_TIME option.

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 that is currently in progress in the event
that the DHCPv6 server determines that it cannot continue processing
an Active Leasequery 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 the
OPTION_LQ_BASE_TIME option in the message. This MUST be the last
message on that connection, and once the message has been
transmitted, the server MUST close the connection.

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

8.4.1. Processing Replies from a Request Containing an
        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 send information
   about real-time binding updates over the same connection. Thus, the
   catch-up phase can 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. In such case, when 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
   the OPTION_STATUS_CODE option) and LEASEQUERY-DATA messages will
   begin to 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.

   The updates sent by the DHCPv6 server during the catch-up phase are
   not in the order that the binding data was updated. Therefore, 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 (and used in a subsequent Active
   Leasequery's query-start-time option), 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 transmits a LEASEQUERY-DATA message with the DHCPv6
   status code of CatchUpComplete (or a LEASEQUERY-REPLY message with a

DHCPv6 status code of DataMissing, as discussed above). 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 8.4, 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 (during transmission,
the aging algorithm causes [some of] 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 that 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 can be
one or more messages concerning a single binding during the catch-up
phase of processing.

Bindings can change multiple times while the requestor is not
connected (that is, during the time from the OPTION_LQ_START_TIME to
the present). The requestor will only receive information about the
current state of the binding, not information about each state change
that occurred during the period from the OPTION_LQ_START_TIME to the
present.

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 an OPTION_LQ_START_TIME option with the same value as the

OPTION_LQ_START_TIME option previously included in the Active
Leasequery 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.

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.

8.5. Processing Time Values in Leasequery Messages

Active or Bulk Leasequery requests can 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; 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 options should the
requestor need to ever issue an Active or Bulk Leasequery message
using these options as part of a later query, since these options
require a time in the context of the DHCPv6 server.

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

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

8.6. Examples

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

8.6.1. Query Failure

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

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

8.6.2. Data Missing on Server

   This example illustrates the message flows in case the 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

8.6.3. Successful Query

   This example illustrates the message flows in case of successful
   query processing by the 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 the DHCPv6 server wants to abort
an in-process request and terminate the connection due to some
reason, it sends LEASEQUERY-DONE with an error code present in the
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)
```

8.7. 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 that supports Active Leasequery MUST support DHCPv6
Bulk Leasequery [RFC5460] along with the updates mentioned in this
document.

9.1. Accepting Connections

DHCPv6 servers that implement DHCPv6 Active Leasequery listen for
incoming TCP connections. The approach used in accepting the
requestor's connection is the same as specified in DHCPv6 Bulk
Leasequery [RFC5460], with the exception that support for Active
Leasequery MUST NOT be enabled by default and MUST require an
explicit configuration step to be performed before it will operate.

DHCPv6 servers SHOULD be able to operate in either insecure or secure
mode. This MAY be a mode that is administratively controlled, where
the server will require a TLS connection to operate or will only
operate without a TLS connection. In either case, operation in
insecure mode MUST NOT be the default, even if operation in secure
mode is not supported. Operation in insecure mode MUST always
require an explicit configuration step, separate from the
configuration step required to enable support for Active Leasequery.

When operating in insecure mode, the DHCPv6 server simply waits for
the requestor to send the Active Leasequery request after the
establishment of a TCP connection. If it receives a STARTTLS
message, it MUST respond with a REPLY [RFC3315] message with a DHCPv6
status code of TLSConnectionRefused.

When operating in secure mode, DHCPv6 servers MUST support TLS
[RFC5246] to protect the integrity and privacy of the data
transmitted over the TCP connection. When operating in secure mode,
DHCPv6 servers MUST be configurable with regard to which requestors
they will communicate. The certificate presented by a requestor when
initiating the TLS connection is used to distinguish between
acceptable and unacceptable requestors.

When operating in secure mode, the DHCPv6 server MUST begin to
negotiate a TLS connection with a requestor who asks for one and MUST
close the TCP connections that are not secured with TLS or for which
the requestor's certificate is deemed unacceptable. The
recommendations in [RFC7525] SHOULD be followed when negotiating a
TLS connection.

A requestor will request a TLS connection by sending a STARTTLS as
the first message over a newly created TCP connection. If the DHCPv6
server supports TLS connections and has not been configured to not
allow them on this link, the DHCPv6 server MUST respond to this
STARTTLS message by sending a REPLY [RFC3315] message without a
DHCPv6 status code back to the requestor. This indicates to the
requestor that the DHCPv6 server will support the negotiation of a
TLS connection over this existing TCP connection.

If for some reason the DHCPv6 server cannot support a TLS connection
or has been configured to not support a TLS connection, then it
SHOULD send a REPLY message with a DHCPv6 status code of
TLSConnectionRefused back to the requestor.

In the event that the DHCPv6 server sends a REPLY message without a
DHCPv6 status code option included (which indicates success), the
requestor is supposed to initiate a TLS handshake [RFC5246] (see
Section 8.2). During the TLS handshake, the DHCPv6 server MUST
validate the requestor's digital certificate. In addition, the
digital certificate presented by the requestor is used to decide if
this requestor is allowed to perform an Active Leasequery. If this
requestor's certificate is deemed unacceptable, the server MUST abort
the creation of the TLS connection.

All TLS connections established between a requestor and a DHCPv6
server for the purposes of supporting Active Leasequery MUST be
mutually authenticated.

If the TLS handshake is not successful in creating a TLS connection,
the server MUST close the TCP connection.

9.2. Rejecting Connections

Servers that do not implement DHCPv6 Active and Bulk Leasequery
SHOULD NOT listen for incoming TCP connections for these requests.

If the DHCPv6 server supporting Bulk Leasequery and not Active
Leasequery receives an Active Leasequery request, it SHOULD send a
LEASEQUERY-REPLY with a DHCPv6 status code of NotSupported. It
SHOULD close the TCP connection after this error is signaled.

9.3. 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 terminate the TCP connection
after ACTIVE_LQ_SEND_TIMEOUT. This timeout governs for how long the
DHCPv6 server is prepared to wait for the requestor to read and
process enough information to unblock the TCP 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 SHOULD close the TCP
connection.

If the DHCPv6 server encounters an error during the 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 SHOULD send
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
SHOULD send 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], with the exception that a server implementing Active
Leasequery SHOULD be able to be configured to prevent specific data
items from being sent to the requestor even if these data items were
requested in the OPTION_ORO option.

Some servers can be configured to respond to a DHCPv6 Leasequery
[RFC5007] and DHCPv6 Bulk Leasequery [RFC5460] for an IPv6 binding
that is reserved in such a way that it appears that the IPv6 binding
is leased to the DHCP client for which it is reserved. These servers
SHOULD also respond to an Active Leasequery request with the same
information as they would to a Bulk Leasequery request when they
first determine that the IPv6 binding is reserved to a DHCP client.

If an Active Leasequery or Bulk Leasequery request contains the
OPTION_LQ_BASE_TIME option code present in OPTION_ORO, the DHCPv6
server MUST include the OPTION_LQ_BASE_TIME option in every reply for
this request. The value for the base-time option is the 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 the 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 that 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 closes the connection.

9.4. Multiple or Parallel Queries

Every Active Leasequery request MUST be made on a single TCP
connection where there is no other request active at the time the
request is made.

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 in parallel would
be possible and reasonable. In case of parallel Active and Bulk
Leasequeries, the requestor MUST use different TCP connections.

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 requestors seeking service.

9.5. 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.

   The server SHOULD limit the number of connections it maintains and
   SHOULD 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 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 can also exhaust a server's pool of available
   connections.

   When operating in secure mode, TLS [RFC5246] is used to secure the
   connection. The recommendations in [RFC7525] SHOULD be followed when
   negotiating a TLS connection.

   Servers SHOULD offer configuration parameters to limit the sources of
   incoming connections through validation and use of the digital
   certificates presented to create a TLS connection. They SHOULD also
   limit the number of accepted connections and limit the period of time
   during which an idle connection will be left open.

   The data acquired by using an Active Leasequery is subject to the
   same potential abuse as the data held by the DHCPv6 server from which
   it was acquired and SHOULD be secured by mechanisms as strong as
   those used for the data held by that DHCPv6 server. The data
   acquired by using an Active Leasequery SHOULD be deleted as soon as
   possible after the use for which it was acquired has passed.

   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 has assigned new DHCPv6 option codes in the "Option Codes"
   registry maintained at <http://www.iana.org/assignments/
   dhcpv6-parameters>:

      OPTION_LQ_BASE_TIME (100)

      OPTION_LQ_START_TIME (101)

      OPTION_LQ_END_TIME (102)

   IANA has assigned new values in the DHCPv6 "Status Codes" registry
   maintained at <http://www.iana.org/assignments/dhcpv6-parameters>:

      DataMissing (12)

      CatchUpComplete (13)

      NotSupported (14)

      TLSConnectionRefused (15)

   IANA has assigned values for the following new DHCPv6 message types
   in the "Message Types" registry maintained at
   <http://www.iana.org/assignments/dhcpv6-parameters>:

      ACTIVELEASEQUERY (22)

      STARTTLS (23)

12. References

12.1. Normative References

   [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>.

   [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>.

   [RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng,
              "DHCPv6 Leasequery", RFC 5007, DOI 10.17487/RFC5007,
              September 2007, <http://www.rfc-editor.org/info/rfc5007>.

   [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
              (TLS) Protocol Version 1.2", RFC 5246,
              DOI 10.17487/RFC5246, August 2008,
              <http://www.rfc-editor.org/info/rfc5246>.

   [RFC5460] Stapp, M., "DHCPv6 Bulk Leasequery", RFC 5460,
              DOI 10.17487/RFC5460, February 2009,
              <http://www.rfc-editor.org/info/rfc5460>.

   [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre,
              "Recommendations for Secure Use of Transport Layer
              Security (TLS) and Datagram Transport Layer Security
              (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May
              2015, <http://www.rfc-editor.org/info/rfc7525>.

12.2. Informative References

   [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>.

   [RFC7414] Duke, M., Braden, R., Eddy, W., Blanton, E., and A.
              Zimmermann, "A Roadmap for Transmission Control Protocol
              (TCP) Specification Documents", RFC 7414,
              DOI 10.17487/RFC7414, February 2015,
              <http://www.rfc-editor.org/info/rfc7414>.

Authors' Addresses

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

   Phone: +91 80 4426-7372
   Email: draghuva@cisco.com

   Kim Kinnear
   Cisco Systems, Inc.
   1414 Massachusetts Avenue
   Boxborough, Massachusetts 01719
   United States

   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 80 4426-7346
   Email: dkukrety@cisco.com