Internet Engineering Task Force (IETF)
Request for Comments: 7724
Updates: 6926
Category: Standards Track
ISSN: 2070-1721

K. Kinnear
M. Stapp
B. Volz
Cisco Systems
N. Russell
Staples

December 2015

Active DHCPv4 Lease Query

Abstract

The Dynamic Host Configuration Protocol for IPv4 (DHCPv4) has been extended with a Leasequery capability that allows a requestor to request information about DHCPv4 bindings (RFC 4388). That mechanism is limited to queries for individual bindings. In some situations, individual binding queries may not be efficient, or even possible. In addition, continuous update of an external requestor with Leasequery data is sometimes desired. This document expands on the DHCPv4 Leasequery protocol, and allows for active transfer of near real-time DHCPv4 binding information data via TCP. This document updates RFC 6926, "DHCPv4 Bulk Leasequery".

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/rfc7724.

Table of Contents

1. Introduction

The DHCPv4 Leasequery capability [RFC4388] extends the basic DHCPv4 capability [RFC2131] [RFC2132] to allow an external entity to query a DHCPv4 server to recover lease state information about a particular IPv4 address or client in near real-time.

Continuous update of an external requestor with Leasequery data is sometimes desired. These requestors need to keep up with the current binding activity of the DHCPv4 server. Keeping up with these binding activities is termed "active" leasequery.

The DHCPv4 Bulk Leasequery [RFC6926] capability can be used to recover useful information from a DHCPv4 server when some external entity starts up. This entity could be one that is directly involved in the DHCPv4 client-server transactions (e.g., a relay agent), or it could be an external process that needs information present in the DHCPv4 server's lease state database.

The Active Leasequery capability documented here is designed to allow an entity not directly involved in DHCPv4 client-server transactions to nevertheless keep current with the state of the DHCPv4 lease state information in real-time.

This document updates DHCPv4 Bulk Leasequery [RFC6926] in that it specifies the DHCPv4 server must close the TCP connection if it receives a DHCPv4 message that is not allowed over the TCP connection (for example, DHCPDISCOVER, DHCPLEASEQUERY). See Section 8.1.1.

2. Terminology

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

This document uses the following terms:

- "Active Leasequery"

  Keeping up to date in real-time (or near real-time) with DHCPv4 binding activity.

- "binding"

  The information that a DHCPv4 server keeps regarding the relationship between a DHCPv4 client and an IPv4 address. This includes the identity of the DHCPv4 client and the expiration time, if any, of any lease that client has on a particular IPv4 address.

- "Bulk Leasequery"

  Requesting and receiving the information about all or some of the existing DHCPv4 binding information in an efficient manner, as defined by [RFC6926].

- "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.

- "catch-up information"

  If a DHCPv4 Active Leasequery requestor sends in a query-start-time option in a DHCPACTIVELEASEQUERY message, the DHCPv4 server will attempt to send the requestor the information that changed since the time specified in the query-start-time option. The binding information sent to satisfy this request is the catch-up information.

- "catch-up phase"

  The period while the catch-up information is being sent is the catch-up phase.

- "clock skew"

  The difference between the absolute time on a DHCPv4 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 DHCPv4 server, a more accurate value is derived from continuously examining the instantaneous value developed from each packet received from a DHCPv4 server and using it to make small adjustments to the existing value held in the requestor.

- "DHCPv4 client"

  A DHCPv4 client is an IPv4 node using DHCP to obtain configuration parameters such as a network address.

- "DHCPv4 relay agent"

  A DHCPv4 relay agent is a third-party agent that transfers BOOTP and DHCPv4 messages between clients and servers residing on different subnets, per [RFC951] and [RFC1542].

- "DHCPv4 server"

  A DHCPv4 server is an IPv4 node that returns configuration parameters to DHCPv4 clients.

- "insecure mode"

  When operating in insecure mode, the TCP connection between the requestor and DHCPv4 server is not protected in any way. In addition, the identity of the requestor is not validated by the server nor is the identity of the server validated by the requestor.

- "MAC address"

  In the context of a DHCP message, a Media Access Control (MAC) address consists of the fields: hardware type "htype", hardware length "hlen", and client hardware address "chaddr".

- "requestor"

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

- "secure mode"

  When operating in secure mode, the TCP connection between the requestor and the DHCPv4 server is protected by TLS [RFC5246]. In addition, the requestor uses the certificates exchanged between it and the DHCPv4 server while setting up the TLS connection to validate the identity of the server. The DHCPv4 server also uses these certificates to validate the identity of the requestor.

3. Protocol Overview

The Active Leasequery mechanism is modeled on the existing individual Leasequery protocol in [RFC4388] as well as related work on DHCPv4 Bulk Leasequery [RFC6926]; most differences arise from the long-term nature of the TCP [RFC7414] connection required for Active Leasequery. In addition, a DHCPv4 server that supports Active Leasequery must support Bulk Leasequery [RFC6926] as well. See Section 8.

An Active Leasequery requestor opens a TCP connection to a DHCPv4 Server, using the DHCPv4 port 67. Note that this implies that the Leasequery requestor has the server IPv4 address(es) available via configuration or some other means, and that it has unicast IP

reachability to the DHCPv4 server. The message framing for TCP is discussed in Section 5.1. No relaying for Active Leasequery is specified.

After establishing a connection, the requestor sends an DHCPACTIVELEASEQUERY message over the connection. In response, the server sends updates to the requestor using DHCPLEASEACTIVE and DHCPLEASEUNASSIGNED messages that are extensions of these messages as defined in [RFC4388] and [RFC6926]. This response procedure is similar to the procedure specified in [RFC6926], 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 the long-lived connection. Additionally, the Active Leasequery server should provide a mechanism to control which data is allowed to be included in the messages sent to the requestor. See Section 8.2.

Since [RFC6926] did not specify what to do with an unknown message type received over the DHCP TCP connection, system administrators SHOULD NOT allow a DHCPACTIVELEASEQUERY message to be sent over a DHCP TCP connection to a DHCPv4 server that does not support Active Leasequery.

Active Leasequery is designed to provide continuous updates of DHCPv4 binding activity to an external entity.

Active Leasequery has features that allow this external entity to lose its connection and then reconnect and receive the latest information concerning any IPv4 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 that works when the requestor is always connected to the DHCPv4 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 DHCPv4 server from which it is receiving updates and then becomes reconnected to that server.

Central to this approach is the concept that, 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 DHCPv4 server will determine whether or not data was missed while the Active Leasequery requestor was not connected.

The DHCP server processing the Active Leasequery request MAY limit the amount of data saved, and methods exist for the DHCPv4 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 [RFC6926] to recover information not available through an Active Leasequery.

DHCPv4 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 DHCPv4 Active Leasequery requestor whose connection was temporarily interrupted.

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 DHCPv4 server was lost for a longer time than the DHCPv4 server would keep track of the specific changes to the IPv4 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 time, clock skew, data source, and other items discussed in the Bulk Leasequery specification [RFC6926] 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 response to the requestor even if they were requested by inclusion in the dhcp-parameter-request-list option.

4. Interaction between Active Leasequery and Bulk Leasequery

Active Leasequery is an extension of the Bulk Leasequery protocol [RFC6926]. The contents of messages returned to an Active Leasequery requestor are identical to those defined for the Bulk Leasequery protocol.

Applications that employ Active Leasequery to keep a database up to date with respect to the DHCPv4 server’s lease state database should use an initial Bulk Leasequery to bring their database into

equivalence with that of the DHCPv4 server, and then use Active Leasequery to keep that database current with respect to the DHCPv4 server's lease state database.

There are several differences between the Active and Bulk Leasequery protocols. Active Leasequery defines only one qualifier (the query-start-time) and no query types, while Bulk Leasequery defines several query types and qualifiers. An Active Leasequery connection sends all available updates to the requestor.

An Active Leasequery connection does not ever "complete", though the DHCPv4 server can close the connection for a variety of reasons associated with some sort of exception condition.

5. Message and Option Definitions

5.1. Message Framing for TCP

The use of TCP for the Active Leasequery protocol permits one or more DHCPv4 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 framing technique used for Bulk Leasequery [RFC6926] is used for Active Leasequery.

When using TLS to secure a connection [RFC5246], the message framing for TLS uses the same format as that used for TCP. One DHCP message is carried in one TLS record.

5.2. New or Changed Options

The existing messages DHCPLEASEUNASSIGNED and DHCPLEASEACTIVE are used as the value of the dhcp-message-type option to indicate an IPv4 address that is currently not leased or is currently leased to a DHCPv4 client, respectively.

All of the message types and options defined for Bulk Leasequery [RFC6926] are also used by Active Leasequery. In addition, new message types and option types are defined for Active Leasequery, as described below.

5.2.1. dhcp-message-type

The message type option (option 53) from [RFC2132] requires additional values. The values of these message types are shown below in an extension of the table from Section 9.6 of [RFC2132]:

Value	Message Type
16	DHCPACTIVELEASEQUERY
17	DHCPLEASEQUERYSTATUS
18	DHCPTLS

5.2.2. dhcp-status-code

The dhcp-status-code option defined in [RFC6926] allows greater detail to be returned regarding the status of a DHCP request. While specified in the Bulk Leasequery document, this DHCPv4 option is also used in Active Leasequery.

This option has two possible scopes when used with Active Leasequery, depending on the context in which it appears. It refers to the information in a single leasequery reply if the value of the dhcp-message-type is DHCPLEASEACTIVE, DHCPLEASEUNASSIGNED, or DHCPTLS. It refers to the message stream related to an entire request if the value of the dhcp-message-type is DHCPLEASEQUERYSTATUS.

Additional status codes defined for support of Active Leasequery are:

Name	Status-Code	Description
DataMissing	5	Indicates that IPv4 binding information requested is not available.
ConnectionActive	6	Indicates that this connection remains active.
CatchUpComplete	7	Indicates that this Active Leasequery connection has completed sending all of the saved data requested.
TLSConnectionRefused	8	Indicates that a TLS connection is not allowed.

A dhcp-status-code option MAY appear in the options field of a DHCP message. If the dhcp-status-code option does not appear, it is assumed that the operation was successful. The dhcp-status-code option SHOULD NOT appear in a message that is successful unless it is needed to convey some text message along with the Success status code.

5.3. Connection and Transmission Parameters

Active Leasequery uses the same port configuration as DHCPv4 Bulk Leasequery [RFC6926]. It also uses other transmission parameters (BULK_LQ_DATA_TIMEOUT and BULK_LQ_MAX_CONNS) as defined in [RFC6926].

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 s	Active Leasequery receive timeout
ACTIVE_LQ_SEND_TIMEOUT	120 s	Active Leasequery send timeout
ACTIVE_LQ_IDLE_TIMEOUT	60 s	Active Leasequery idle timeout

6. Information Communicated by Active Leasequery

While the information communicated by a Bulk Leasequery [RFC6926] is taken directly from the DHCPv4 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 DHCPv4 server’s lease state database.

This is of significance, because if the Active Leasequery requestor runs slowly or the requestor disconnects from the DHCPv4 server and then reconnects with a query-start-time (signaling a catch-up operation), the information communicated to the Active Leasequery requestor is only the most current information from the DHCPv4 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 DHCPv4 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 DHCPv4 server is waiting to send information down the connection, when the connection becomes available to be written, the DHCPv4 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 DHCPv4 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 never miss any changes in the binding.

7. Requestor Behavior

7.1. General Processing

A requestor attempts to establish a TCP connection to a DHCPv4 server in order to initiate a 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 5.3.

If an Active Leasequery is terminated prematurely by a DHCPLEASEQUERYDONE with a dhcp-message status-code 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 5.3.

Messages from the DHCPv4 server come as multiple responses to a single DHCPACTIVELEASEQUERY message. Thus, each DHCPACTIVELEASEQUERY or DHCPBULKLEASEQUERY request must have an xid (transaction-id) unique on the connection on which it is sent (see Section 7.3), and all of the messages that come as a response to it contain the same xid as the request.

Only one DHCPACTIVELEASEQUERY is allowed on any one TCP connection at a time. Parallel DHCPACTIVELEASEQUERY requests on the same TCP connection are not allowed.

7.2. Initiating a Connection

A requestor SHOULD be able to operate in either insecure or secure mode. See Section 9. This MAY be a feature that is administratively controlled.

When operating in insecure mode, the requestor sends a DHCPACTIVELEASEQUERY request 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] apply when negotiating this connection.

A requestor requests the establishment of a TLS connection by sending the DHCPTLS message to the DHCPv4 server as the first message over the TCP connection. The DHCPTLS message SHOULD be sent without any options. This message indicates to the DHCPv4 server that a TLS connection over this TCP connection is desired. There are four possibilities after the requestor sends the DHCPTLS message to the DHCPV4 server:

1. No response from the DHCPv4 server.

2. The DHCPv4 server closes the TCP connection after it receives the DHCPTLS message.

3. DHCPv4 server responds with a DHCPTLS message with a dhcp-status-code of TLSConnectionRefused.

4. DHCPv4 server responds with DHCPTLS message with no dhcp-status-code, indicating success.

In any of the first three possibilities, the DHCPv4 server can be assumed to not support TLS. In this case, the requestor MUST close the connection.

In the final possibility, where the DHCPv4 server has responded with a DHCPTLS message with no dhcp-status-code in response to the requestor’s DHCPTLS 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 DHCPv4 server's digital certificates.

If the handshake exchange yields a functioning TLS connection, then the requestor SHOULD transmit a DHCPACTIVELEASEQUERY message over that TLS connection and use that TLS connection for all further interactions in which it engages with the DHCPv4 server over this TCP connection.

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

7.3. Forming an Active Leasequery

The Active Leasequery is designed to create a long-lived connection between the requestor and the DHCPv4 server processing the active query. The DHCPv4 server SHOULD send binding information back across this connection with minimal delay after it learns of the binding information. It will learn about the bindings either because it makes the bindings itself or because it has received information about a binding from another server.

An Active Leasequery is a DHCPv4 request with a dhcp-message-type of DHCPACTIVELEASEQUERY. The DHCPv4 request MUST NOT have a ciaddr, a chaddr, or a dhcp-client-identifier. The DHCPv4 request MUST have an xid (transaction-id) unique on the connection on which it is sent. The DHCPv4 request SHOULD have a dhcp-parameter-request-list to inform the DHCPv4 server which DHCPv4 options are of interest to the requestor sending the DHCPACTIVELEASEQUERY message.

An important capability of the Active Leasequery is that the requestor can 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 DHCPv4 server processing a previous Active Leasequery.

This capability is enabled by the transmission of a 4-octet 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 DHCPv4 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 DHCPv4 server, the requestor should place this

highest base-time value into a query-start-time option in the new DHCPACTIVELEASEQUERY request. (See Sections 6.2.5 and 7.2 of [RFC6926] for information on the query-start-time option.)

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

If the requestor doesn't wish to request an update of information missed when it was not connected to the DHCPv4 server, then it does not include the query-start-time option in the DHCPACTIVELEASEQUERY 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.

7.4. Processing Active Replies

The Requestor attempts to read a DHCPv4 leasequery reply message from the TCP connection.

Note that the connection resulting from accepting a DHCPACTIVELEASEQUERY request may be long-lived and may not have data transferring continuously during its lifetime. Therefore, the DHCPv4 server SHOULD send a DHCPLEASEQUERYSTATUS message with a dhcp-status-code of ConnectionActive 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 the connection is idle (i.e., the server has no binding data to send). During normal binding data exchange, receiving DHCPLEASEACTIVE or DHCPLEASEUNASSIGNED messages by the requestor itself signifies that the 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 DHCPv4 server to send messages. Thus, ACTIVE_LQ_RCV_TIMEOUT controls how sensitive the requestor is to be to delays by the DHCPv4 server in sending updates or DHCPLEASEQUERYSTATUS messages.

If the stream of replies becomes blocked with no messages being received, the Requestor SHOULD terminate the connection after ACTIVE_LQ_RCV_TIMEOUT, and MAY begin retry processing if configured to do so.

A successful query that is returning binding data MUST include a non-zero ciaddr. It may also include a non-zero chaddr, htype, and hlen as well as additional options. If there are additional bindings to be returned, they will be carried in additional Active Leasequery messages.

Any requestor of an Active Leasequery operation MUST be prepared to receive multiple copies of the binding information for a particular IPv4 address. See the Bulk Leasequery document [RFC6926] for information on how to deal with this situation.

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 DHCPACTIVELEASEQUERY message from a single DHCPv4 server.

A DHCPACTIVELEASEQUERY has two regimes -- during the catch-up phase, if any, and after any catch-up phase. If the DHCPACTIVELASEQUERY request had a query-start-time, then the DHCPACTIVELEASEQUERY starts out in the catch-up phase. See Section 7.4.1 for information on processing during the catch-up phase, as well as how to determine when the catch-up phase is complete.

After the catch-up phase, or during the entire series of messages received as the response to a DHCPACTIVELEASEQUERY request with no query-start-time (and therefore no catch-up phase), the 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 DHCPv4 server) can be used in a subsequent DHCPACTIVELEASEQUERY message's query-start-time or in a DHCPBULKLEASEQUERY message's query-start-time, if one is required, after a loss of the Active Leasequery connection.

The DHCPLEASEQUERYSTATUS message MAY unilaterally terminate a successful DHCPACTIVELEASEQUERY request that is currently in progress in the event that the DHCPv4 server determines that it cannot continue processing a DHCPACTIVELEASEQUERY request. For example, when a server is requested to shut down, it SHOULD send a DHCPLEASEQUERYSTATUS message with a dhcp-status-code of QueryTerminated and include in the message a base-time. This MUST be the last message on that connection, and once the message has been transmitted, the server MUST close the connection.

After receiving DHCPLEASEQUERYSTATUS with a QueryTerminated status from a server, the Requestor MAY close the TCP connection to that server.

The DHCPv4 Leasequery protocol uses the associated-ip option as an indicator that multiple bindings were present in response to a single client-based query. For Active Leasequery, client-based queries are not supported, and so the associated-ip option is not used and MUST NOT be present in replies.

7.4.1. Processing Replies from a Request Containing a query-start-time

If the DHCPACTIVELEASEQUERY was requested with a query-start-time, the DHCPv4 server will attempt to send information about all bindings that changed since the time specified in the query-start-time. This is the catch-up phase of the DHCPACTIVELEASEQUERY processing. The DHCPv4 server MAY also begin immediate updates over the same connection of real-time binding information changes. Thus, the catch-up phase can run in parallel with the normal updates generated by the DHCPACTIVELEASEQUERY request.

A DHCPv4 server MAY keep only a limited amount of time-ordered information available to respond to a DHCPACTIVELEASEQUERY request containing a query-start-time. Thus, it is possible that the time specified in the query-start-time represents a time not covered by the time-ordered information kept by the DHCPv4 server. In such case, when there is not enough data saved in the DHCPv4 server to satisfy the request specified by the query-start-time option, the DHCPv4 server will reply immediately with a DHCPLEASEQUERYSTATUS message with a dhcp-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 DHCPACTIVELEASEQUERY request.

If there is enough data saved to satisfy the request, then DHCPLEASEACTIVE and DHCPLEASEUNASSIGNED messages will begin arrive from the DHCPv4 server. Some of these messages will be related to the query-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 DHCPv4 server. In general, there is no way to determine the source of each message.

The updates sent by the DHCPv4 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 DHCPv4 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 DHCPACTIVELEASEQUERY were to terminate prior to the completion of the catch-up phase.

The requestor will know that the catch-up phase is complete because the DHCPv4 server will transmit a DHCPLEASEQUERYSTATUS message with the dhcp-status-code of CatchUpComplete (or, as discussed above, DataMissing). Once this message is transmitted, all additional DHCPLEASEACTIVE and DHCPLEASEUNASSIGNED messages will relate to real-time ("new") binding changes in the DHCPv4 server.

As discussed in Section 6.3, the requestor SHOULD keep track of the latest base-time option value received over a particular connection, to be used in a subsequent DHCPACTIVELEASEQUERY 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 DHCPLEASEQUERYDONE 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 DHCPv4 server to begin to satisfy the request implied by the query-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 was very slow, and the aging algorithm caused the saved data to become unavailable), the DHCPv4 server will terminate the catch-up phase of processing immediately by sending a DHCPLEASEQUERYSTATUS message with a dhcp-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 query-start-time and the present is replicated in an Active Leasequery reply message. See Section 6. 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 query-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. 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 query-start-time to the present.

If the DHCPLEASEQUERYSTATUS message containing a dhcp-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 the bindings in the DHCPv4 server, then the requestor SHOULD issue a DHCPBULKLEASEQUERY request to recover the information missing from

its database. This DHCPBULKLEASEQUERY should include a query-start-time option, set to the same value as the query-start-time option previously included in the DHCPACTIVELEASEQUERY responses from the DHCPv4 server, and a query-end-time option equal to the base-time option returned by the DHCPv4 server in the DHCPLEASEQUERYSTATUS message with the dhcp-status-code of DataMissing.

Typically, the requestor would have one connection open to a DHCPv4 server for a DHCPACTIVELEASEQUERY request and possibly one additional connection open for a DHCPBULKLEASEQUERY request to the same DHCPv4 server to fill in the data that might have been missed prior to the initiation of the DHCPACTIVELEASEQUERY. The Bulk Leasequery connection would typically run to completion and be closed, leaving one Active Leasequery connection open to a single DHCPv4 server.

7.5. Closing Connections

The Requestor or DHCPv4 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.

8. Server Behavior

A DHCPv4 server that supports Active Leasequery MUST support Bulk Leasequery [RFC6926] as well.

8.1. Accepting Connections

DHCPv4 servers that implement DHCPv4 Active Leasequery listen for incoming TCP connections. The approach used in accepting the requestor’s connection is the same as specified in DHCPv4 Bulk Leasequery [RFC6926], 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.

DHCPv4 servers SHOULD be able to operate in either insecure or secure mode. See Section 9. 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 DHCPv4 server simply waits for the requestor to send the Active Leasequery after the establishment of TCP connection. If it receives a DHCPTLS message, it will respond with TLSConnectionRefused in a DHCPTLS message.

When operating in secure mode, DHCPv4 servers MUST support TLS [RFC5246] to protect the integrity and privacy of the data transmitted over the TCP connection. When operating in secure mode, DHCPv4 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, a DHCPv4 server MUST begin to negotiate a TLS connection with a requestor who asks for one, and MUST close TCP connections that are not secured with TLS or for which the requestor's certificate is deemed unacceptable. The recommendations in [RFC7525] apply when negotiating a TLS connection.

A requestor will request a TLS connection by sending a DHCPTLS as the first message over a newly created TCP connection. If the DHCPv4 server supports TLS connections and has not been configured to not allow them on this link, the DHCPv4 server MUST respond to this DHCPTLS message by sending a DHCPTLS message with no dhcp-status-code back to the requestor. This indicates to the requestor that the DHCPv4 server will support the negotiation of a TLS connection over this existing TCP connection.

If a connection is to be rejected because of a limitation of the number of open connections, the TCP connection itself should be rejected, or the subsequent ACTIVELEASEQUERY message should be rejected. Capacity-related rejections SHOULD NOT affect the response to the DHCPTLS message.

Any options appearing in a DHCPTLS message received by a DHCPv4 server SHOULD be ignored. This is a "SHOULD" instead of a "MUST" in order to allow use of the DHCPTLS message in later documents, possibly with the use of options, without requiring those documents to update this document.

If for some reason the DHCPv4 server cannot support or has been configured to not support a TLS connection, then it sends a DHCPTLS message with a dhcp-status-code of TLSConnectionRefused back to the requestor.

In the event that the DHCPv4 server sends a DHCPTLS message with no dhcp-status-code option included (which indicates success), the requestor is supposed to initiate a TLS handshake [RFC5246] (see

Section 7.2). During the TLS handshake, the DHCPv4 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 DHCPv4 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.

If the TCP connection becomes blocked while the server is accepting a connection or reading a query, it SHOULD terminate the connection after a BULK_LQ_DATA_TIMEOUT. We make this recommendation to allow servers to control the period of time they are willing to wait before abandoning an inactive connection, independent of the TCP implementations they may be using.

8.1.1. Update to RFC 6926

In an update to the DHCPv4 Bulk Leasequery protocol [RFC6926] (which didn't discuss this situation explicitly), if the DHCPv4 server receives a DHCPv4 message containing a dhcp-message-type option with a value that is not supported over a TCP connection, it MUST close the TCP connection.

8.2. Replying to an Active Leasequery

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 how long the DHCPv4 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 DHCPv4 server SHOULD close the TCP connection.

If the DHCPv4 server encounters an error during processing of the DHCPACTIVELEASEQUERY message, either during initial processing or later during the message processing, it SHOULD send a DHCPLEASEQUERYSTATUS containing an error code of some kind in a dhcp-status-code option. It SHOULD close the connection after this error is signaled.

Every reply to a DHCPACTIVELEASEQUERY request MUST contain the information specified in replies to a DHCPBULKLEASEQUERY request [RFC6926], 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 dhcp-parameter-request-list option.

Some servers can be configured to respond to a DHCPv4 Leasequery [RFC4388] or a DHCPBULKLEASEQUERY [RFC6926] for an IPv4 binding that is reserved in such a way that it appears that the IPv4 binding is leased to the DHCP client for which it is reserved. These servers SHOULD also respond to a DHCPACTIVELEASEQUERY request with the same information as they would to a DHCPBULKLEASEQUERY request when they first determine that the IPv4 binding is reserved to a DHCP client.

If a DHCPACTIVELEASEQUERY request contains a query-start-time option, it indicates that the requestor would like the DHCPv4 server to send it not only messages that correspond to DHCPv4 binding activity that occurs subsequent to the receipt of the DHCPLEASEACTIVE request, but also messages that correspond to DHCPv4 binding activity that occurred prior to the DHCPACTIVELEASEQUERY request.

If a query-end-time option appears in a DHCPACTIVELEASEQUERY the DHCPv4 server should send a DHCPLEASEQUERYSTATUS message with a dhcp-status-code of MalformedQuery and terminate the connection.

In order to implement a meaningful response to this query, the DHCPv4 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 a DHCPACTIVELEASEQUERY request containing a query-start-time option, the DHCPv4 server can respond with replies for all binding activity occurring on that query-start-time or later times.

These replies based on the query-start-time MAY be interleaved with the messages generated due to current binding activity.

Once the transmission of the DHCPv4 Leasequery messages associated with the query-start-time option are complete, a DHCPLEASEQUERYSTATUS message MUST be sent with a dhcp-status-code value of CatchUpComplete.

The DHCPv4 server SHOULD keep track of previous binding activity. It SHOULD limit the amount of previous binding activity it keeps track of. The DHCPv4 server MAY choose to only do this in the event that it has received at least one DHCPACTIVELEASEQUERY request in the past, as to do so will almost certainly entail some utilization of resources that would be wasted if there are no DHCPACTIVELEASEQUERY

requestors for this DHCPv4 server. The DHCPv4 server SHOULD make the amount of previous binding activity it retains configurable. There is no requirement on the DHCPv4 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 DHCPACTIVELEASEQUERY request yielding a DHCPLEASEQUERYSTATUS message, such as a server shutdown, there will be no DHCPLEASEQUERYSTATUS message at the conclusion of the DHCPACTIVELEASEQUERY processing because that processing will not conclude but will continue until either the requestor or the server closes the connection.

While the form of the data being sent by a DHCPACTIVELEASEQUERY is essentially the same as that being sent by a DHCPBULKLEASEQUERY, the reasons for sending information differs considerably between these two capabilities. In the DHCPBULKLEASEQUERY context, the entire contents of the lease state database (subject to the constraints of the various query options) are returned to the requestor. In the DHCPACTIVELEASEQUERY context, changes to the lease state database are returned to the requestor essentially as they happen. For instance, when an IPv4 binding transitions from the leased state to some other state, the DHCPACTIVELEASEQUERY will send a DHCPLEASEUNASSIGNED packet with information regarding that binding. The server may then entirely forget about that IPv4 binding (or not), but it is important to tell the DHCPACTIVELEASEQUERY requestor that a binding has transitioned away from the leased state.

The relationship between the time that the server replies to a DHCP client request and the time that the DHCP server sends a reply to a DHCPACTIVELEASEQUERY message is a matter of implementation (and thus not defined by this document). However, the server SHOULD NOT delay responding to the DHCP client in order to transmit a reply to a DHCPACTIVELEASEQUERY message, and the server SHOULD send the reply to the DHCPACTIVELASEQUERY message as soon as possible after responding to the client.

8.3. 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. Note that this is different than what was allowed in Section 7.7 of [RFC6926] for Bulk Leasequery requests.

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 Leasequery requests, the requestor MUST use different 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.

8.4. Closing Connections

The server MAY end communication by sending a DHCPLEASEQUERYSTATUS message and then immediately closing the TCP connection. Alternatively, the server MAY retain the connection and wait for additional queries from the requestor. The server SHOULD limit the number of connections it maintains and SHOULD close idle connections to enforce the limit.

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 dhcp-status-code option in a DHCPLEASEQUERYSTATUS message. If the server detects that the requestor end has been closed, the server MUST close its end of the connection.

9. Security Considerations

The Security Considerations section of [RFC2131] details the general threats to DHCPv4. The DHCPv4 Leasequery specification [RFC4388] describes recommendations for the Leasequery protocol, especially with regard to relayed LEASEQUERY messages, mitigation of packet-flooding 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 DHCPv4 server’s available TCP connection resources can compromise the ability of legitimate clients 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.

Two modes of operation exist for this protocol, insecure mode and secure mode. These two modes exist because there are essentially two models of use for this protocol. In one model, the requestor of an Active Leasequery is connected to the Internet in an arbitrary location, and the information transmitted needs to be protected

during transmission. In addition, the identities of both requestor and server need to be verified. For this model of use, the secure mode is appropriate.

The other model of use is where the requestor of the Active Leasequery resides in a network element that is essentially "next to" the element containing the DHCP server, and both of these elements are inside a protected environment. For this model, the insecure mode is sufficient since there are other, more global, protections in place to protect this information.

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

Operating in insecure mode (see Section 8.1) does not provide any way to validate the authorization of requestors of a DHCPV4 Active Leasequery request.

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 DHCPv4 server from which it was acquired and SHOULD be secured by mechanisms as strong as those used for the data held by that DHCPv4 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.

Servers that implement the Bulk Leasequery protocol [RFC6926] but do not implement the Active Leasequery protocol SHOULD implement the update to [RFC6926] discussed in Section 8.1.1.

10. IANA Considerations

IANA has assigned the following new DHCP message types from the registry "DHCP Message Type 53 Values" maintained at <http://www.iana.org/assignments/bootp-dhcp-parameters>:

1. A dhcp-message-type of 16 for DHCPACTIVELEASEQUERY.

2. A dhcp-message-type of 17 for DHCPLEASEQUERYSTATUS.

3. A dhcp-message-type of 18 for DHCPTLS.

IANA has assigned the following new DHCP status codes from the registry "DHCP Status Code Type 151 Values" maintained at <http://www.iana.org/assignments/bootp-dhcp-parameters>:

Name	Status-Code
DataMissing	5
ConnectionActive	6
CatchUpComplete	7
TLSConnectionRefused	8

11. References

11.1. 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>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <http://www.rfc-editor.org/info/rfc2131>.

[RFC4388] Woundy, R. and K. Kinnear, "Dynamic Host Configuration Protocol (DHCP) Leasequery", RFC 4388, DOI 10.17487/RFC4388, February 2006, <http://www.rfc-editor.org/info/rfc4388>.

[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>.

[RFC6926] Kinnear, K., Stapp, M., Desetti, R., Joshi, B., Russell, N., Kurapati, P., and B. Volz, "DHCPv4 Bulk Leasequery", RFC 6926, DOI 10.17487/RFC6926, April 2013, <http://www.rfc-editor.org/info/rfc6926>.

[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>.

11.2. Informative References

[RFC951] Croft, W. and J. Gilmore, "Bootstrap Protocol", RFC 951, DOI 10.17487/RFC0951, September 1985, <http://www.rfc-editor.org/info/rfc951>.

[RFC1542] Wimer, W., "Clarifications and Extensions for the Bootstrap Protocol", RFC 1542, DOI 10.17487/RFC1542, October 1993, <http://www.rfc-editor.org/info/rfc1542>.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <http://www.rfc-editor.org/info/rfc2132>.

[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>.

Acknowledgments

The ideas in this document came in part from work in DHCPv6 and DHCPv4 Bulk Leasequery as well as from in depth discussions between the authors. Useful review comments by Ted Lemon, Scott Bradner, Francis Dupont, and Stephen Farrell on drafts for DHCPv6 Active Leasequery were also included in this draft. Brian Haberman’s review brought this document into much closer alignment with DHCPv6 Active Leasequery. Additional reviews by Alissa Cooper, Spencer Dawkins, Christer Holmberg, and Ben Campbell added clarity to this document.

Authors' Addresses

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

Email: kkinnear@cisco.com

Mark Stapp
Cisco Systems, Inc.
1414 Massachusetts Ave
Boxborough, MA 01719
United States

Email: mjs@cisco.com

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

Email: volz@cisco.com

Neil Russell
Staples
500 Staples Drive
Framingham, MA 01702
United States

Email: neil.e.russell@gmail.com