Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 5, 2014

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

June 3, 2014

Active DHCPv4 Lease Query
draft-ietf-dhc-dhcpv4-active-leasequery-01.txt

Abstract

The Dynamic Host Configuration Protocol for IPv4 (DHCPv4) has been extended with a Leasequery capability that allows a client to request information about DHCPv4 bindings. 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 client with Leasequery data is sometimes desired. This document expands on the DHCPv4 Leasequery protocol, and allows for active transfer of near real-time DHCPv4 address binding information data via TCP.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

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

This Internet-Draft will expire on December 5, 2014.

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.

Requirements exist for external entities to keep up to date on the correspondence between DHCPv4 clients and the IPv4 addresses for which they have leases. These requirements often stem from regulatory requirements placed on service providers by governmental agencies.

These entities need to keep up with the current IPv4 address binding activity of the DHCPv4 server. Keeping up with address binding activity 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 which is directly involved in the DHCPv4 client - server transactions (e.g., a relay agent), or it could be an external process which 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.

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:

- "address 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.

- "Active Leasequery"

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

- "Bulk Leasequery"

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

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

  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 address binding information sent to satisfy this request is the catch-up information, and the period while it is being sent is the catch-up phase.

- "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 [RFC0951] and [RFC1542].

- "DHCPv4 server"

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

- "IP address 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.

- "MAC address"

  In the context of a DHCP message, a 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.

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 connection required for Active Leasequery. In addition, a DHCPv4 server which supports Active Leasequery MUST support Bulk Leasequery [RFC6926] as well.

An Active Leasequery client opens a TCP connection to a DHCPv4 Server, using the DHCPv4 port 67. Note that this implies that the Leasequery client 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 discusssed in Section 5.1. No relaying for Active Leasequery is specified.

After establishing a connection, the client sends an DHCPACTIVELEASEQUERY message over the connection. In response, the server sends updates to the requestor using DHCPLEASEACTIVE and DHCPLEASEUNASSIGNED messages which are extensions of these messages as defined in [RFC4388] and [RFC6926].

Active Leasequery is designed to provide continuous updates of DHCPv4 IPv4 address binding activity to an external entity.

Active Leasequery has features which allow this external entity to lose its connection and then reconnect and receive the latest information concerning any IPv4 addresses changed while it was not connected.

These capabilities are designed to allow the Active Leasequery requestor to efficiently become current with respect to the lease state database after it has been restarted or the machine on which it is running has been reinitialized. It is easy to define a protocol which works when the requestor is always connected to the 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, 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 a Active Leasequery connection, but will typically choose to keep data corresponding to some recent activity available for subsequent queries by a DHCPv4 Active Leasequery client 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 address 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 address 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.

4. Interaction Between Active Leasequery and Bulk Leasequery

Active Leasequery can be seen as an extension of the Bulk Leasequery protocol [RFC6926]. The contents of packets returned to an Active

Leasequery requestor are identical to that defined for the Bulk Leasequery protocol.

Applications which employ Active Leasequery to keep a database up to date with respect to the DHCPv4 server's lease state database will usually 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 may drop 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 at a time. 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.

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 which is currently not leased or 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
TBD1	DHCPACTIVELEASEQUERY
TBD2	DHCPLEASEQUERYSTATUS

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 or DHCPLEASEUNASSIGNED. 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	TBD3	Indicates that IPv4 address binding information requested is not available.
ConnectionActive	TBD4	Indicates that this connection remains active.
CatchUpComplete	TBD5	Indicates that this Active Leasequery connection has completed sending all of the saved data requested.

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 which is successful unless it is needed to convey some text message along with the Success status code.

5.3. Connection and Transmission Parameters

DHCPv4 servers that support Active Leasequery SHOULD listen for incoming TCP connections on port 67. Implementations MAY offer to make the incoming port configurable, but port 67 MUST be the default. Requestors SHOULD make TCP connections to port 67, and MAY offer to make the destination server port configurable.

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
BULK_LQ_DATA_TIMEOUT	*	Bulk Leasequery data timeout
BULK_LQ_MAX_CONNS	*	Max Bulk Leasequery TCP connections
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

* See Section 6.3 of [RFC6926] for specific default values.

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 which is currently associated with a particular IPv4 address 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 an IPv4 address 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 IPv4 address 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 an IPv4 address 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 client to IPv4 address binding.

7. Requestor Behavior

7.1. Connecting and 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.

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.

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, and all of the messages which come as a response to it all contain the same xid as the request. It is the xid which allows the data-streams of two or more different DHCPACTIVELEASEQUERY or DHCPBULKLEASEQUERY requests to be demultiplexed by the requestor.

A requestor MAY send a DHCPACTIVELEASEQUERY request to a DHCPv4 server and immediately close the transmission side of its TCP

connection, and then read the resulting response messages from the DHCPv4 server. This is not required, and the usual approach is to leave both sides of the TCP connection up until at least the conclusion of the Active Leasequery.

7.2. 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 will send IPv4 address binding information back across this connection with minimal delay after it learns of the binding information. It will learn about IPv4 address bindings either because it makes the bindings itself or because it has received information about a binding from another server.

To form the Active query, a DHCPv4 request is constructed with a dhcp-message-type of DHCPACTIVELEASEQUERY. This DHCPv4 request MUST NOT have a ciaddr, a chaddr, or a dhcp-client-identifier. The DHCPv4 request MUST contain a transaction-id, and that transaction-id MUST be locally unique to the TCP connection to the DHCPv4 server. 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 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 IPv4 address 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.

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.

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 will typically 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 will 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.)

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 be prepared to 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.

7.3. Processing Active Replies

The Requestor attempts to read a DHCPv4 leasequery reply message from the TCP connection. If the stream of replies becomes blocked, the Requestor SHOULD be prepared to terminate the connection after ACTIVE_LQ_RCV_TIMEOUT, and MAY begin retry processing if configured to do so.

Note that a DHCPACTIVELEASEQUERY request specifically requests the DHCPv4 server to create a long-lived connection which may not have data transferring continuously during its lifetime. Therefore the DHCPv4 server will 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. 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.

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 IPv4 address 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. During the catch-up phase (if one exists), the data returned in the base-time option in a DHCPLEASEACTIVE or DHCPLEASEUNASSIGNED message may appear to be ordered, but the most recent change in the lease state data being returned is not related to the base-time option value in the messages. Another way to say this is that the ordering of the updates sent by the DHCPv4 server during the catch-up phase is independent of the ordering in the changes in the lease state data. The base-time option from messages during this phase MUST NOT be saved and used in a subsequent DHCPACTIVELEASEQUERY message's query-start-time option as it does not represent the extent of progress of the catch-up activity.

After the catch-up phase, or during the entire series of messages received as the response to 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 which 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 SHOULD be the last message on that connection, and once the message has been transmistted, the server should 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.

7.3.1. 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 IPv4 address 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 IPv4 address binding information changes. Thus, the catch-up phase may 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. If this should occur, and 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 IPv4 address binding changes taking place in the DHCPv4 server. In general, there is no way to determine the source each message.

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 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. Once this message is transmitted, all additional DHCPLEASEACTIVE and DHCPLEASEUNASSIGNED messages will relate to real-time ("new") IPv4 address 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 is very slow, and the aging algorithm causes 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 IPv4 address during the period from the time specified in the query-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 IPv4 address which 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 IPv4 address will contain the state at the current time, and there may be one or more messages concerning a single IPv4 address during the catch-up phase of processing.

If an IPv4 address changed state multiple times during the time that the requestor was not connected (that is, during the time from the query-start-time and the present), then only the current IPv4 address binding information will be sent during the catch-up phase. Thus, the requestor MUST NOT assume that every intermediate state change that occurred during the period from the query-start-time to the present will be represented by an individual Leasequery message.

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 IPv4 address 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, set to be the same as its query-start-time previously included in the DHCPACTIVELEASEQUERY responses from the DHCPv4 server, and a query-end-time equal to the base-time returned by the

DHCPv4 server in the DHCPLEASEQUERYSTATUS message with the dhcp-status-code of DataMissing.

In the event that the requestor receives a DHCPLEASEQUERYSTATUS message with a dhcp-status-code of DataMissing, it is a reasonable assumption that it is interested in keeping its database up to date with respect to the DHCPv4 server's internal IPv4 address binding database or it would not have included the query-start-time in the DHCPACTIVELEASEQUERY message.

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. Alternatively, both requests could be issued over a single connection.

7.4. 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 client 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 which supports Active Leasequery MUST support Bulk Leasequery [RFC6926] as well.

8.1. Accepting Connections

Servers that implement DHCPv4 Active Leasequery listen for incoming TCP connections. Port numbers are discussed in Section 5.3. Servers MUST be able to limit the number of currently accepted and active connections. The value BULK_LQ_MAX_CONNS MUST be the default; implementations MAY permit the value to be configurable. Connections SHOULD be accepted and, if the number of connections is over BULK_LQ_MAX_CONNS, they SHOULD be closed immediately.

Servers MAY restrict Active Leasequery connections and DHCPACTIVELEASEQUERY messages to certain clients. Connections not from permitted clients SHOULD be closed immediately, to avoid server connection resource exhaustion.

If the TCP connection becomes blocked while the server is accepting a connection or reading a query, it SHOULD be prepared to 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.

8.2. Replying to an Active Leasequery

If the connection becomes blocked while the server is attempting to send reply messages, the server SHOULD be prepared to terminate the TCP connection after ACTIVE_LQ_SEND_TIMEOUT. This timeout governs how much congestion the DHCPv4 server is prepared to tolerate over any Active Leasequery connection. The default is two minutes, which means that if more than two minutes goes by without the requestor reading enough information to unblock the TCP connection, the DHCPv4 server will drop 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].

Some servers can be configured to respond to a DHCPv4 Leasequery [RFC4388] or a DHCPBULKLEASEQUERY [RFC6926] for an IPv4 address which is reserved in such a way that it appears that the IPv4 address 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 address 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 address binding activity that occurs subsequent to the receipt of the DHCPLEASEACTIVE request, but also messages that correspond to DHCPv4 address 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 address 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 address 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 IPv4 address 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, but is not required to, keep track of a limited amount of previous address binding activity and associate it with base-time values. 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 which would be wasted if there are no DHCPACTIVELEASEQUERY clients for this DHCPv4 server. The DHCPv4 server SHOULD make the amount of previous address 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 client or the server drops 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 address transitions from the leased state to some other state, the DHCPACTIVELEASEQUERY will send a DHCPLEASEUNASSIGNED packet with information regarding that IPv4 address. The server may then entirely forget about that IPv4 address (or not), but it is

important to tell the DHCPACTIVELEASEQUERY client that an IPv4 address 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 something 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

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

Typically, a requestor of a Active Leasequery would not need to send a second Active Leasequery while the first is still active. However, sending an Active Leasequery and a Bulk Leasequery over the same connection would be possible and reasonable.

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 close its end of the TCP connection after sending its last message, a DHCPLEASEQUERYSTATUS message in response to a query. Alternatively, the server MAY retain the connection and wait for additional queries from the client. The server SHOULD be prepared to limit the number of connections it maintains, and SHOULD be prepared to close idle connections to enforce the limit.

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 clients by using the QueryTerminated status code in the dhcp-status-code option in a DHCPLEASEQUERYSTATUS message. If the server detects that the client end has been closed,

the server MUST close its end of the connection after it has finished processing any outstanding requests.

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 clients, and use of IPsec [RFC4301].

This capability SHOULD be disabled by default.

The use of TCP introduces some additional concerns. Attacks that attempt to exhaust the DHCPv4 server's available TCP connection resources, such as SYN flooding attacks, can compromise the ability of legitimate clients to receive service. Malicious clients who succeed in establishing connections, but who then send invalid queries, partial queries, or no queries at all also can exhaust a server's pool of available connections. We recommend that servers offer configuration to limit the sources of incoming connections, that they limit the number of accepted connections and the number of in-process queries from any one connection, and that they limit the period of time during which an idle connection will be left open.

10. IANA Considerations

IANA is requested to assign 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 TBD1 for DHCPACTIVELEASEQUERY.

2. A dhcp-message-type of TBD2 for DHCPLEASEQUERYSTATUS.

IANA is requested to assign 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	TBD3
ConnectionActive	TBD4
CatchUpComplete	TBD5

11. Acknowledgements

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.

12. References

12.1. Normative References

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

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4388] Woundy, R. and K. Kinnear, "Dynamic Host Configuration Protocol (DHCP) Leasequery", RFC 4388, February 2006.

[RFC6926] Kinnear, K., Stapp, M., Desetti, R., Joshi, B., Russell, N., Kurapati, P., and B. Volz, "DHCPv4 Bulk Leasequery", RFC 6926, April 2013.

12.2. Informative References

[RFC0951] Croft, B. and J. Gilmore, "Bootstrap Protocol", RFC 951, September 1985.

[RFC1542] Wimer, W., "Clarifications and Extensions for the Bootstrap Protocol", RFC 1542, October 1993.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

Authors' Addresses

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

Email: kkinnear@cisco.com

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

Email: mjs@cisco.com

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

Email: volz@cisco.com

Neil Russell
Staples
500 Staples Drive
Framingham, MA 01702
USA

Email: neil.e.russell@gmail.com