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

K. Kinnear
M. Stapp
Cisco Systems, Inc.
R. Desetti
B. Joshi
Infosys Ltd.
N. Russell
Sea Street Technologies Inc.
P. Kurapati
Juniper Networks
B. Volz
Cisco Systems, Inc.

April 2013

DHCPv4 Bulk Leasequery

Abstract

The Dynamic Host Configuration Protocol for IPv4 (DHCPv4) Leasequery protocol allows a requestor to request information about DHCPv4 bindings. This protocol is limited to queries for individual bindings. In some situations, individual binding queries may not be efficient or even possible. This document extends the DHCPv4 Leasequery protocol to allow for bulk transfer of DHCPv4 address binding data via TCP.

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/rfc6926.

Table of Contents

1. Introduction 4
2. Terminology 5
3. Design Goals 8
3.1. Information Acquisition before Data Starts 8
3.2. Lessen Need for Caching and Negative Caching 8
3.3. Antispoofing in 'Fast Path' 8
3.4. Minimize Data Transmission 9
4. Protocol Overview 9
5. Interaction between UDP Leasequery and Bulk Leasequery 11
6. Message and Option Definitions 12
6.1. Message Framing for TCP 12
6.2. New or Changed Options 13
6.3. Connection and Transmission Parameters 20
7. Requestor Behavior 21
7.1. Connecting and General Processing 21
7.2. Forming a Bulk Leasequery 21
7.3. Processing Bulk Replies 23
7.4. Processing Time Values in Leasequery Messages 25
7.5. Querying Multiple Servers 26
7.6. Making Sense out of Multiple Responses concerning a Single IPv4 Address 26
7.7. Multiple Queries to a Single Server over One Connection 27
7.8. Closing Connections 28
8. Server Behavior 29
8.1. Accepting Connections 29
8.2. Replying to a Bulk Leasequery 29
8.3. Building a Single Reply for Bulk Leasequery 33
8.4. Multiple or Parallel Queries 34
8.5. Closing Connections 35
9. Security Considerations 35
10. IANA Considerations 37
11. Acknowledgements 38
12. References 38
12.1. Normative References 38
12.2. Informative References 39

1. Introduction

DHCPv4 [RFC2131] [RFC2132] specifies a protocol for the assignment of IPv4 address and configuration information to IPv4 nodes. DHCPv4 servers maintain authoritative binding information.

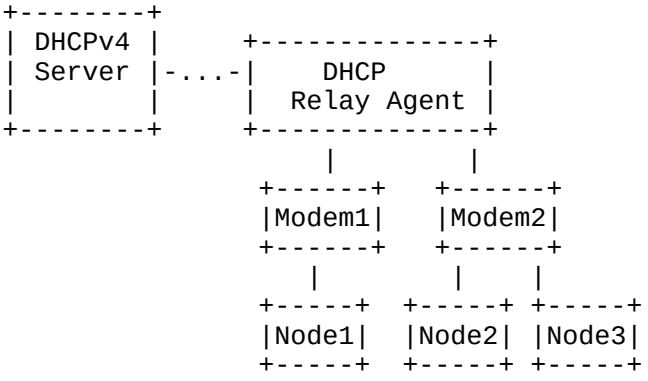


Figure 1: Example DHCPv4 Configuration

DHCPv4 relay agents receive DHCPv4 messages and frequently append a Relay Agent Information option [RFC3046] before relaying them to the configured DHCPv4 servers (see Figure 1). In this process, some relay agents also glean lease information sent by the server and cache it locally. This information is used for a variety of purposes. Two examples are prevention of spoofing attempts from the DHCPv4 clients and installation of routes. When a relay agent reboots, this information is frequently lost.

The DHCPv4 Leasequery capability [RFC4388] extends the basic DHCPv4 capability to allow an external entity, such as a relay agent, to query a DHCPv4 server to rapidly recover lease state information about a particular IP address or client.

The existing query types in Leasequery are typically data driven; the relay agent initiates the Leasequery when it receives data traffic from or to the client. This approach may not scale well when there are thousands of clients connected to the relay agent or when the relay agent has a need to rebuild its internal data store prior to processing traffic in one direction or another.

Some applications require the ability to query the server without waiting for traffic from or to clients. This query capability, in turn, requires an underlying transport more suitable to the bulk transmission of data.

This document extends the DHCPv4 Leasequery protocol [RFC4388] to add support for queries that address these additional requirements. There may be many thousands of DHCPv4 bindings returned as the result of a single request, so TCP [RFC4614] is specified for efficiency of data transfer. We define several additional query types, each of which can return multiple responses, in order to meet a variety of requirements.

2. Terminology

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

This document uses the following terms:

- "absolute time"

  Absolute time is a 32-bit quantity containing the number of seconds since January 1, 1970.

- "access concentrator"

  An access concentrator is a router or switch at the broadband access provider's edge of a public broadband access network. This document assumes that the access concentrator includes the DHCPv4 relay agent functionality, for example, a CMTS (Cable Modem Termination System) in a cable environment or a DSLAM (Digital Subscriber Line Access Multiplexer) in a DSL environment.

- "active binding"

  An IP address with an active binding refers to an IP address that is currently associated with a DHCPv4 client where that DHCPv4 client has the right to use the IP address.

- "Bulk Leasequery"

  Bulk Leasequery involves requesting and receiving the existing DHCPv4 address binding information in an efficient manner.

- "clock skew"

  The clock skew for a Bulk Leasequery connection is the difference between the absolute time on a DHCPv4 server and the absolute time on the system where a requestor of a Bulk Leasequery is executing. It is not absolutely constant but is likely to vary only slowly.

It is possible that, when both systems run NTP, the clock skew is negligible; this is not only acceptable but desired.

While it is easy to think that this can be calculated precisely after one message is received by a requestor from a DHCPv4 server, a more accurate value is derived from continuously examining the instantaneous value developed from each message received from a DHCPv4 server and using it to make small adjustments to the existing value held in the requestor.

- "Default VPN"

  A default VPN indicates that the address being described belongs to the set of addresses not part of any VPN (in other words, the normal address space operated on by DHCP). This includes Special Use IPv4 Addresses as defined in [RFC5735].

- "DHCPv4 client"

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

- "DHCPv4 relay agent"

  A DHCPv4 relay agent is an agent that is neither a DHCPv4 client nor a DHCP server 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 Internet node that returns configuration parameters to DHCPv4 clients.

- "DSLAM"

  DSLAM stands for Digital Subscriber Line Access Multiplexer.

- "downstream"

  Downstream refers to a direction away from the central part of a network and toward the edge. In a DHCPv4 context, this typically refers to a network direction that is away from the DHCPv4 server and toward the DHCPv4 client.

- "Global VPN"

  Global VPN is another name for the default VPN.

- "IP address"

  In this document, the term "IP address" refers to an IPv4 IP address.

- "IP address binding"

  An IP address binding is the information that a DHCPv4 server keeps regarding the relationship between a DHCPv4 client and an IP address. This includes the identity of the DHCPv4 client and the expiration time, if any, of any lease that client has on a particular IP address. In some contexts, this may include information on IP addresses that are currently associated with DHCPv4 clients, and in others, it may also include IP addresses with no current association to a DHCPv4 client.

- "MAC address"

  In the context of a DHCPv4 message, a Media Access Control (MAC) address consists of the fields: hardware type "htype", hardware length "hlen", and client hardware address "chaddr".

- "upstream"

  Upstream refers to a direction toward the central part of a network and away from the edge. In a DHCPv4 context, this typically refers to a network direction that is away from the DHCPv4 client and toward the DHCPv4 server.

- "stable storage"

  Stable storage is used to hold information concerning IP address bindings (among other things) so that this information is not lost in the event of a failure that requires restart of the network element. DHCPv4 servers are typically expected to have high-speed access to stable storage, while relay agents and access concentrators usually do not have access to stable storage, although they may have periodic access to such storage.

- "xid"

  Transaction-id. The term "xid" refers to the DHCPv4 field containing the transaction-id of the message.

3. Design Goals

The goal of this document is to provide a lightweight protocol for an access concentrator or other network element (such as a DHCP relay agent) to retrieve IP address binding information available in the DHCPv4 server. The protocol should also allow an access concentrator or DHCP relay agent to retrieve consolidated IP address binding information for either the entire access concentrator or a single connection/circuit. Throughout the discussion below, everything that applies to an access concentrator also applies to a DHCP relay agent.

3.1. Information Acquisition before Data Starts

The existing data-driven approach required by [RFC4388] means that the Leasequeries can only be performed after an access concentrator receives data. To implement antispoofing, the concentrator must drop messages for each client until it gets lease information from the DHCPv4 server for that client. If an access concentrator finishes the Leasequeries before it starts receiving data, then there is no need to drop legitimate messages. In this way, outage time may be reduced.

3.2. Lessen Need for Caching and Negative Caching

The result of a single Leasequery should be cached, whether that results in a positive or negative cache, in order to remember that the Leasequery was performed. This caching is required to limit the traffic imposed upon a DHCPv4 server by Leasequeries for information already received.

These caches not only consume precious resources, they also need to be managed. Hence, they should be avoided as much as possible. One of the goals of the DHCPv4 Bulk Leasequery is to reduce the need for this sort of caching.

3.3. Antispoofing in 'Fast Path'

If antispoofing is not done in the fast path, it will become a bottleneck and may lead to denial of service of the access concentrator. The Leasequeries should make it possible to do antispoofing in the fast path.

3.4. Minimize Data Transmission

It may be that a network element is able to periodically save its entire list of assigned IP addresses to some form of stable storage. In this case, it will wish to recover all of the updates to this information without duplicating the information it has recovered from its own stable storage.

Bulk Leasequery allows the specification of a query-start-time as well as a query-end-time. Use of query times allows a network element that periodically commits information to stable storage to recover just what it lost since the last commit.

4. Protocol Overview

The DHCPv4 Bulk Leasequery protocol is modeled on the existing individual DHCPv4 Leasequery protocol in [RFC4388] as well as related work on DHCPv6 Bulk Leasequery [RFC5460]. A Bulk Leasequery requestor opens a TCP connection to a DHCPv4 server using the DHCPv4 port 67. Note that this implies that the Leasequery requestor has server IP address(es) available via configuration or some other means and that it has unicast IP reachability to the DHCPv4 server. No relaying of Bulk Leasequery messages is specified.

After establishing a connection, the requestor sends a DHCPBULKLEASEQUERY message over the connection.

The server uses the message type and additional data in the DHCPv4 DHCPBULKLEASEQUERY message to identify any relevant bindings.

In order to support some query types, servers may have to maintain additional data structures or otherwise be able to locate bindings that have been requested by the Leasequery requestor.

Relevant bindings are returned in DHCPv4 messages with either the DHCPLEASEACTIVE message type for an IP address with a currently active lease or, in some situations, a DHCPLEASEUNASSIGNED message type for an IP address that is controlled by the DHCPv4 server but is not actively leased by a DHCPv4 client at the present time.

The Bulk Leasequery protocol is designed to provide an external entity with information concerning existing DHCPv4 IPv4 address bindings managed by the DHCPv4 server. When complete, the DHCPv4 server will send a DHCPLEASEQUERYDONE message. If a connection is lost while processing a Bulk Leasequery, the Bulk Leasequery must be retried as there is no provision for determining the extent of data already received by the requestor for a Bulk Leasequery.

Bulk Leasequery supports queries by MAC address and by Client Identifier in a way similar to [RFC4388]. The Bulk Leasequery protocol also adds several new queries.

- Query by Relay Identifier

  This query asks a server for the bindings associated with a specific relay agent; the relay agent is identified by a Relay Agent Identifier carried in a Relay-ID sub-option [RFC6925]. Relay agents can include this sub-option while relaying messages to DHCPv4 servers. Servers can retain the Relay-ID and associate it with bindings made on behalf of the relay agent's clients. The bindings returned are only those for DHCPv4 clients with a currently active binding.

- Query by Remote ID

  This query asks a server for the bindings associated with a relay agent Remote ID sub-option [RFC3046] value. The bindings returned are only those for DHCPv4 clients with a currently active binding.

- Query for All Configured IP Addresses

  This query asks a server for information concerning all IP addresses configured in that DHCPv4 server by specifying no other type of query. In this case, the bindings returned are for all configured IP addresses, whether or not they contain a currently active binding to a DHCPv4 client, since one point of this type of query is to update an existing database with changes after a particular point in time.

Any of the above queries can be qualified by the specification of a query-start-time or a query-end-time (or both). When these timers are used as qualifiers, they indicate that a binding should be included if it changed on or after the query-start-time and on or before the query-end-time.

In addition, any of the above queries can be qualified by the specification of a VPN-ID option [RFC6607] to select the VPN on which the query should be processed. The VPN-ID option is also extended to allow queries across all available VPNs. In the absence of any VPN-ID option, only the default (global) VPN is used to satisfy the query.

5. Interaction between UDP Leasequery and Bulk Leasequery

Bulk Leasequery can be seen as an extension of the existing UDP Leasequery protocol [RFC4388]. This section clarifies the relationship between the two protocols.

The Bulk Leasequery TCP connection is only designed to handle the DHCPBULKLEASEQUERY request. It is not intended as an alternative DHCPv4 communication option for clients seeking other DHCPv4 services. DHCPv4 address allocation could not be performed over a TCP connection in any case, as a TCP connection requires an IP address and no IPv4 address exists prior to a successful DHCPv4 address allocation exchange. In addition, the existing DHCPv4 UDP transmission regime is implemented in untold millions of devices deployed worldwide, and complicating DHCPv4 services with alternative transmission approaches (even if it were possible) would be worse than any perceived benefit to doing so.

Two of the query types introduced in the UDP Leasequery protocol can be used in the Bulk Leasequery protocol -- Query by MAC address and Query by Client-identifier.

The contents of the reply messages are similar between the existing UDP Leasequery protocol and the Bulk Leasequery protocol, though more information is returned in the Bulk Leasequery messages.

One change in behavior for these existing queries is required when Bulk Leasequery is used. Sections 6.1, 6.4.1, and 6.4.2 of [RFC4388] specify the use of an associated-ip option in DHCPLEASEACTIVE messages in cases where multiple bindings were found. When Bulk Leasequery is used, this mechanism is not necessary; a server returning multiple bindings simply does so directly as specified in this document. The associated-ip option MUST NOT appear in Bulk Leasequery replies.

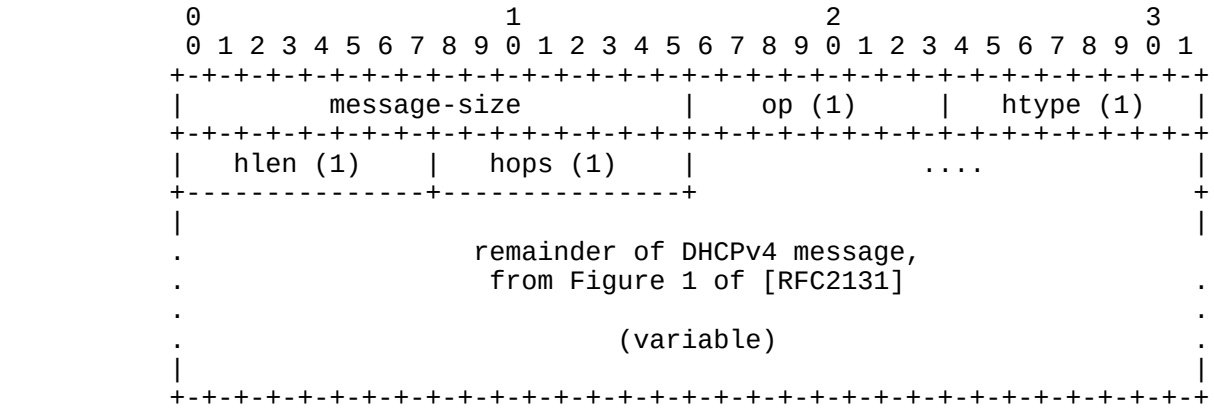
Implementors should note that the TCP message framing defined in Section 6.1 is not compatible with the UDP message format. If a TCP-framed request is sent as a UDP message, it may not be valid, because protocol fields will be offset by the message-size prefix.

6. Message and Option Definitions

6.1. Message Framing for TCP

The use of TCP for the Bulk Leasequery protocol permits multiple messages to be sent from one end of the connection to the other without requiring a request/response paradigm as does UDP DHCPv4 [RFC2131]. The receiver needs to be able to determine the size of each message it receives. Two octets containing the message size in network byte order are prepended to each DHCPv4 message sent on a Bulk Leasequery TCP connection. The two message-size octets 'frame' each DHCPv4 message.

The maximum message size is 65535 octets.



message-size the number of octets in the message that follows, as a 16-bit unsigned integer in network byte order.

All other fields are as specified in DHCPv4 [RFC2131].

Figure 2: Format of a DHCPv4 Message in TCP

The intent in using this format is that code that currently knows how to deal with sending or receiving a message in [RFC2131] format will easily be able to deal with the message contained in the TCP framing.

6.2. New or Changed Options

The existing messages DHCPLEASEUNASSIGNED and DHCPLEASEACTIVE are used as the value of the dhcp-message-type option to indicate an IP address that is currently not leased or currently leased to a DHCPv4 client, respectively [RFC4388].

Additional options have also been defined to enable the Bulk Leasequery protocol to communicate useful information to the requestor.

6.2.1. dhcp-message-type

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

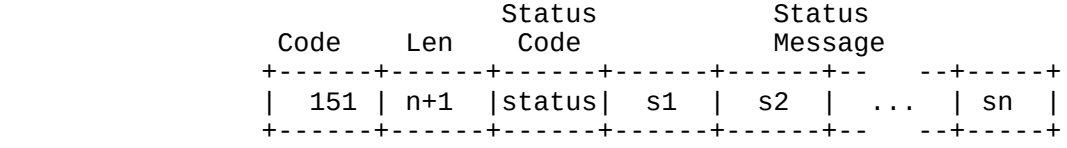
Value	Message Type
14	DHCPBULKLEASEQUERY
15	DHCPLEASEQUERYDONE

6.2.2. status-code

The status-code option allows a machine-readable value to be returned regarding the status of a DHCPBULKLEASEQUERY request.

This option has two possible scopes when used with Bulk 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 DHCPLEASEQUERYDONE.

The code for this option is 151. The length of this option is a minimum of 1 octet.



The status-code is indicated in one octet as defined in the table below. The Status Message is an optional UTF-8-encoded text string suitable for display to an end user. This text string MUST NOT contain a termination character (e.g., a null). The Len field describes the length of the Status Message without any terminator character. Null characters MUST NOT appear in the Status Message string, and it is a protocol violation for them to appear in any position in the Status Message, including at the end.

Name	Status Code	Description
Success	000	Success. Also signaled by absence of a status-code option.
UnspecFail	001	Failure, reason unspecified.
QueryTerminated	002	Indicates that the server is unable to perform a query or has prematurely terminated the query for some reason (which should be communicated in the text message).
MalformedQuery	003	The query was not understood.
NotAllowed	004	The query or request was understood but was not allowed in this context.

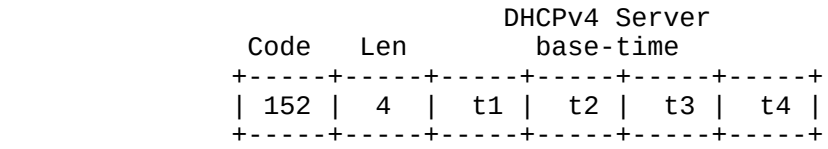
A status-code option MAY appear in the options field of a DHCPv4 message. If the status-code option does not appear, it is assumed that the operation was successful. The status-code option SHOULD NOT appear in a message that is successful unless there is some text string that needs to be communicated to the requestor.

6.2.3. base-time

The base-time option is the current time the message was created to be sent by the DHCPv4 server to the requestor of the Bulk Leasequery. This MUST be an absolute time. All of the other time-based options in the reply message are relative to this time, including the dhcp-lease-time [RFC2132] and client-last-transaction-time [RFC4388]. This time is in the context of the DHCPv4 server that placed this option in a message.

This is an unsigned integer in network byte order.

The code for this option is 152. The length of this option is 4 octets.



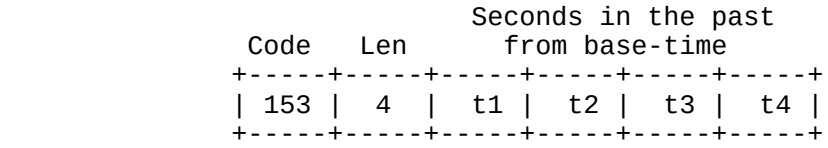
6.2.4. start-time-of-state

The start-time-of-state option allows the receiver to determine the time at which the IP address made the transition into its current state.

This MUST NOT be an absolute time, which is equivalent to saying that this MUST NOT be an absolute number of seconds since January 1, 1970. Instead, this MUST be the unsigned integer number of seconds from the time the IP address transitioned its current state to the time specified in the base-time option in the same message.

This is an unsigned integer in network byte order.

The code for this option is 153. The length of this option is 4 octets.



6.2.5. query-start-time

The query-start-time option specifies a start query time to the DHCPv4 server. If specified, only bindings that have changed on or after the query-start-time should be included in the response to the query.

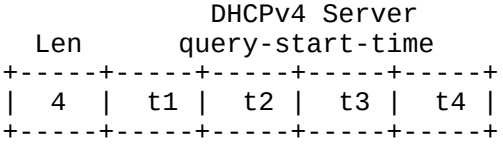
The requestor MUST determine the query-start-time using lease information it has received from the DHCPv4 server. This MUST be an absolute time in the DHCPv4 server's context (see Section 7.4).

Typically (though this is not a requirement), the query-start-time option will contain the value most recently received in a base-time option by the requestor, as this will indicate the last successful communication with the DHCP server.

This MUST be an absolute time.

This is an unsigned integer in network byte order.

The code for this option is 154. The length of this option is 4 octets.



6.2.6. query-end-time

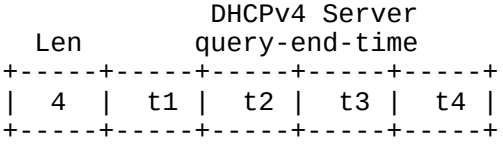
The query-end-time option specifies an end query time to the DHCPv4 server. If specified, only bindings that have changed on or before the query-end-time should be included in the response to the query.

The requestor MUST determine the query-end-time based on lease information it has received from the DHCPv4 server. This MUST be an absolute time in the context of the DHCPv4 server.

In the absence of information to the contrary, the requestor SHOULD assume that the time context of the DHCPv4 server is identical to the time context of the requestor (see Section 7.4).

This is an unsigned integer in network byte order.

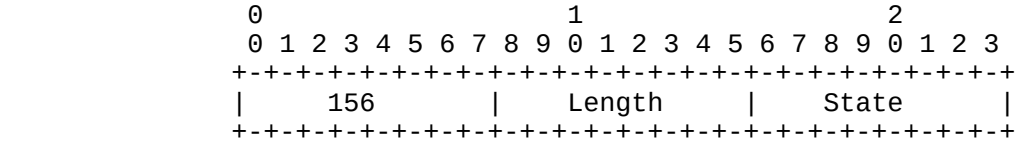
The code for this option is 155. The length of this option is 4 octets.



6.2.7. dhcp-state

The dhcp-state option allows greater detail to be returned than allowed by the DHCPLEASEACTIVE and DHCPLEASEUNASSIGNED message types.

The code for this option is 156. The length of this option is 1 octet.



156 The option code.

Length The option length, 1 octet.

State The state of the IP address.

Value	State	
1	AVAILABLE	Address is available to local DHCPv4 server
2	ACTIVE	Address is assigned to a DHCPv4 client
3	EXPIRED	Lease has expired
4	RELEASED	Lease has been released by DHCPv4 client
5	ABANDONED	Server or client flagged address as unusable
6	RESET	Lease was freed by some external agent
7	REMOTE	Address is available to a remote DHCPv4 server
8	TRANSITIONING	Address is moving between states

Note that some of these states may be transient and may not appear in normal use. A DHCPv4 server MUST implement at least the AVAILABLE and ACTIVE states and SHOULD implement at least the ABANDONED and RESET states.

Note the states AVAILABLE and REMOTE are relative to the current server. An address that is available to the current server should show AVAILABLE on that server, and if another server is involved with that address as well, it should show as REMOTE on that other server.

The dhcp-state option SHOULD contain ACTIVE when it appears in a DHCPLEASEACTIVE message. A DHCPv4 server MAY choose to not send a dhcp-state option in a DHCPLEASEACTIVE message, and a requestor SHOULD assume that the dhcp-state is ACTIVE if no dhcp-state option appears in a DHCPLEASEACTIVE message.

The reference to local and remote relate to possible use in an environment that includes multiple servers cooperating to provide an increased availability solution. In this case, an IP address with the state of AVAILABLE is available to the local server, while one with the state of REMOTE is available to a remote server. Usually, an IP address that is AVAILABLE on one server would be REMOTE on any remote server. The TRANSITIONING state is also likely to be useful in multiple server deployments, where sometimes one server must interlock a state change with one or more other servers. Should a Bulk Leasequery need to send information concerning the state of the IP address during this period, it SHOULD use the TRANSITIONING state, since the IP address is likely to be neither ACTIVE or AVAILABLE.

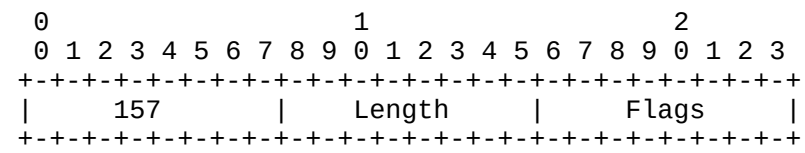
There is no requirement for the state of an IP address to transition in a well-defined way from state to state. To put this another way, you cannot draw a simple state transition graph for the states of an IP address, and the requestor of a Leasequery MUST NOT depend on one certain state always following a particular previous state. While a state transition diagram can be drawn, it would be fully connected and therefore conveys no useful information. Every state can (at times) follow every other state.

6.2.8. data-source

The data-source option contains information about the source of the data in a DHCPLEASEACTIVE or a DHCPLEASEUNASSIGNED message. It SHOULD be used when there are two or more servers that might have information about a particular IP address binding. Frequently, two servers work together to provide an increased availability solution for the DHCPv4 service, and in these cases, both servers will respond to Bulk Leasequery requests for the same IP address. When one server is working with another server and both may respond with information about the same IP address, each server SHOULD return the data-source option with the other information provided about the IP address.

The data contained in this option will allow an external process to better discriminate between the information provided by each of the servers servicing this IPv4 address.

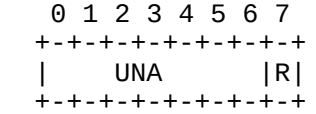
The code for this option is 157. The length of this option is 1 octet.



157 The option code.

Length The option length, 1 octet.

Flags The source information for this message.



R: REMOTE flag

remote = 1
local = 0

UNA: UNASSIGNED

The REMOTE flag is used to indicate where the most recent change of state (or other interesting change) concerning this IPv4 address took place. If the value is local, then the change took place on the server from which this message was transmitted. If the value is remote, then the change took place on some other server and was made known to the server from which this message was transmitted.

If this option was requested and it doesn't appear, the requestor MUST consider that the data-source was local.

Unassigned bits MUST be ignored.

6.2.9. Virtual Subnet Selection Type and Information

All of the (sub-)options defined in [RFC6607] carry identical payloads, consisting of a type and additional VSS (Virtual Subnet Selection) information. The existing table is extended (see below) with a new type 254 to allow specification of a type code that indicates that all VPNs are to be used to process the Bulk Leasequery.

	Type	VSS Information Format
	0	Network Virtual Terminal (NVT) ASCII VPN identifier
	1	RFC 2685 VPN-ID
CHANGED ->	2-253	Unassigned
NEW ->	254	All VPNs (wildcard)
	255	Global, default VPN

6.3. Connection and Transmission Parameters

DHCPv4 servers that support Bulk Leasequery SHOULD listen for incoming TCP connections on the DHCPv4 server 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 Bulk 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 are NOT RECOMMENDED.

Parameter	Default	Description
BULK_LQ_DATA_TIMEOUT	300 secs	Bulk Leasequery data timeout for both client and server (see Sections 7 and 8)
BULK_LQ_MAX_CONNS	10	Max Bulk Leasequery TCP connections at the server side (see Section 8.1)

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 Bulk Leasequery is terminated prematurely by a DHCPLEASEQUERYDONE with a status-code option with a 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 DHCPBULKLEASEQUERY message. Thus, each DHCPBULKLEASEQUERY request MUST have an xid (transaction-id) unique on the connection on which it is sent. All of the messages that come as a response to that message will contain the same xid as the request. The xid allows the data-streams of two different DHCPBULKLEASEQUERY requests to be demultiplexed by the requestor.

7.2. Forming a Bulk Leasequery

Bulk Leasequery is designed to create a connection that will transfer the state of some subset (or possibly all) of the IP address bindings from the DHCPv4 server to the requestor. The DHCPv4 server will send all of the requested IPv4 address bindings across this connection with minimal delay after it receives the request. In this context, "all IP address binding information" means information about all IPv4 addresses configured within the DHCPv4 server that meet the specified query criteria. For some query criteria, this may include IP address binding information for IP addresses that may not now have or ever have had an association with a specific DHCPv4 client.

To form the Bulk query, a DHCPv4 request is constructed with a dhcp-message-type of DHCPBULKLEASEQUERY. The query SHOULD have a dhcp-parameter-request-list to inform the DHCPv4 server which DHCPv4 options are of interest to the requestor sending the DHCPBULKLEASEQUERY message. The dhcp-parameter-request-list in a DHCPBULKLEASEQUERY message SHOULD contain the codes for base-time, dhcp-lease-time, start-time-of-state, and client-last-transaction-time.

A DHCPBULKLEASEQUERY request is constructed of one primary query and optionally one or more qualifiers for it.

The possible primary queries are listed below. Each DHCPBULKLEASEQUERY request MUST contain only one of these primary queries.

- Query by MAC address

  In a Query by MAC address, the chaddr, htype, and hlen of the DHCPv4 packet are filled in with the values requested.

- Query by Client-identifier

  In a Query by Client-identifier, a Client-identifier option containing the requested value is included in the DHCPBULKLEASEQUERY request.

- Query by Remote ID

  In a Query by Remote ID, a Remote ID sub-option containing the requested value is included in the relay-agent-information option of the DHCPBULKLEASEQUERY request.

- Query by Relay-ID

  In a Query by Relay-ID, a Relay-ID sub-option [RFC6925] containing the requested value is included in the relay-agent-information option of the DHCPBULKLEASEQUERY request.

- Query for All Configured IP Addresses

  A Query for All Configured IP addresses is signaled by the absence of any other primary query.

There are three qualifiers that can be applied to any of the above primary queries. These qualifiers can appear individually or together in any combination, but only one of each can appear.

- Query Start Time

  Inclusion of a query-start-time option specifies that only IP address bindings that have changed on or after the time specified in the query-start-time option should be returned.

- Query End Time

  Inclusion of a query-end-time option specifies that only IP address bindings that have changed on or before the time specified in the query-end-time option should be returned.

- o VPN-ID

  If no VPN-ID option appears in the DHCPBULKLEASEQUERY, the default (global) VPN is searched to satisfy the query specified by the DHCPBULKLEASEQUERY. Using the VPN-ID option [RFC6607] allows the requestor to specify a single VPN other than the default VPN. In addition, the VPN-ID option has been extended as part of this document to allow specification that all configured VPNs be searched in order to satisfy the query specified in the DHCPBULKLEASEQUERY.

  In all cases, any message returned from a DHCPBULKLEASEQUERY request containing information about an IP address for other than the default (global) VPN MUST contain a VPN-ID option in the message.

Use of the query-start-time or the query-end-time options or both can serve to reduce the amount of data transferred over the TCP connection by a considerable amount. Note that the times specified in the query-start-time or query-end-time options are absolute times, not durations offset from "now".

The TCP connection may become blocked or stop being writable while the requestor is sending its query. Should this happen, the implementation's behavior is controlled by the current value of BULK_LQ_DATA_TIMEOUT. The default value is given elsewhere in this document, and this value may be overridden by local configuration of the operator.

If this situation is detected, the requestor SHOULD start a timer using the current value of BULK_LQ_DATA_TIMEOUT. If that timer expires, the requestor SHOULD terminate the connection. This timer is completely independent of any TCP timeout established by the TCP protocol connection.

7.3. Processing Bulk Replies

The requestor attempts to read a DHCPv4 Leasequery reply message from the TCP connection.

The TCP connection may stop delivering reply data (i.e., the connection stops being readable). Should this happen, the implementation's behavior is controlled by the current value of BULK_LQ_DATA_TIMEOUT. The default value is given elsewhere in this document, and this value may be overridden by local configuration of the operator.

If this situation is detected, the requestor SHOULD start a timer using the current value of BULK_LQ_DATA_TIMEOUT. If that timer expires, the requestor SHOULD terminate the connection.

A single Bulk Leasequery can, and usually will, result in a large number of replies. The requestor MUST be prepared to receive more than one reply with an xid matching a single DHCPBULKLEASEQUERY message from a single DHCPv4 server. If the xid in the received message does not match an outstanding DHCPBULKLEASEQUERY message, the requestor MUST close the TCP connection.

If the requestor receives more data than it can process, it can simply abort the connection and try again with a more specific request. It can also simply read the TCP connection more slowly and match the rate at which it can digest the information returned in the Bulk Leasequery packets with the rate at which it reads those packets from the TCP connection.

The DHCPv4 server MUST send a server-identifier option (option 54) in the first response to any DHCPBULKLEASEQUERY message. The DHCPv4 server SHOULD NOT send server-identifier options in subsequent responses to that DHCPBULKLEASEQUERY message. The requestor MUST cache the server-identifier option from the first response and apply it to any subsequent responses.

The response messages generated by a DHCPBULKLEASEQUERY request are:

- DHCPLEASEACTIVE

  A Bulk Leasequery will generate DHCPLEASEACTIVE messages containing binding data for bound IP addresses that match the specified query criteria. The IP address that is bound to a DHCPv4 client will appear in the ciaddr field of the DHCPLEASEACTIVE message. The message may contain a non-zero chaddr, htype, hlen, and possibly additional options.

- DHCPLEASEUNASSIGNED

  Some queries will also generate DHCPLEASEUNASSIGNED messages for IP addresses that match the query criteria. These messages indicate that the IP address is managed by the DHCPv4 server but is not currently bound to any DHCPv4 client. The IP address to which this message refers will appear in the ciaddr field of the DHCPLEASEUNASSIGNED message. A DHCPLEASEUNASSGINED message MAY also contain information about the last DHCPv4 client that was bound to this IP address. The message may contain a non-zero chaddr, htype, hlen, and possibly additional options in this case.

- DHCPLEASEQUERYDONE

  A response of DHCPLEASEQUERYDONE indicates that the server has completed its response to the query and that no more messages will be sent in response to the DHCPBULKLEASEQUERY. More details will sometimes be available in the received status-code option in the DHCPLEASEQUERYDONE message. If there is no status-code option in the DHCPLEASEQUERYDONE message, then the query completed successfully.

  Note that a query that returned no data, that is, a DHCPBULKLEASEQUERY request followed by a DHCPLEASEQUERYDONE response, is considered a successful query in that no errors occurred during the processing. It is not considered an error to have no information to return to a DHCPBULKLEASEQUERY request.

The DHCPLEASEUNKNOWN message MUST NOT appear in a response to a Bulk Leasequery.

The requestor MUST NOT assume that there is any inherent order in the IP address binding information that is sent in response to a DHCPBULKLEASEQUERY. While the base-time will tend to increase monotonically (as it is the current time on the DHCPv4 server), the actual time that any IP address binding information changed is unrelated to the base-time.

The DHCPLEASEQUERYDONE message always ends a successful DHCPBULKLEASEQUERY request and any unsuccessful DHCPBULKLEASEQUERY requests not terminated by a dropped connection. After receiving a DHCPLEASEQUERYDONE from a server, the requestor MAY close the TCP connection to that server if no other DHCPBULKLEASEQUERY is outstanding on that TCP connection.

The DHCPv4 Leasequery protocol [RFC4388] uses the associated-ip option as an indicator that multiple bindings were present in response to a single DHCPv4 client-based query. For Bulk Leasequery, a separate message is returned for each binding, so the associated-ip option is not used.

7.4. Processing Time Values in Leasequery Messages

Bulk Leasequery requests may be made to a DHCPv4 server whose absolute time may not be synchronized with the local time of the requestor. Thus, there are at least two time contexts in even the simplest Bulk Leasequery response, and in the situation where multiple DHCPv4 servers are queried, the situation becomes even more complex.

If the requestor of a Bulk Leasequery is saving the data returned in some form, it has a requirement to store a variety of time values; some of these will be time in the context of the requestor, and some will be time in the context of the DHCPv4 server.

When receiving a DHCPLEASEACTIVE or DHCPLEASEUNASSIGNED message from the DHCPv4 server, the message will contain a base-time option. The time contained in this base-time option is in the context of the DHCPv4 server. As such, it is an ideal time to save and use as input to a DHCPBULKLEASEQUERY in the query-start-time or query-end-time options, should the requestor ever need to issue a DHCPBULKLEASEQUERY message using those options as part of a later query, since those options require a time in the context of the DHCPv4 server.

In addition to saving the base-time for possible future use in a query-start-time or query-end-time option, the base-time is used as part of the conversion of the other times in the Leasequery message to values that are meaningful in the context of the requestor. These other time values are specified as a offset (duration) from the base-time value and not as an absolute time.

In systems whose clocks are synchronized, perhaps using NTP, the clock skew will usually be zero.

7.5. Querying Multiple Servers

A Bulk Leasequery requestor MAY be configured to attempt to connect to and query from multiple DHCPv4 servers in parallel. The DHCPv4 Leasequery specification [RFC4388] includes a discussion about reconciling binding data received from multiple DHCPv4 servers.

In addition, the algorithm in Section 7.6 should be used.

7.6. Making Sense out of Multiple Responses concerning a Single IPv4 Address

Any requestor of an Bulk Leasequery MUST be prepared for multiple responses to arrive for a particular IPv4 address from multiple different DHCPv4 servers. The following algorithm SHOULD be used to decide if the information just received is more up to date (i.e., better) than the best existing information. In the discussion below, the information that is received from a DHCPv4 server about a particular IPv4 address is termed a "record". The times used in the algorithm below SHOULD have been converted into the requestor’s context, and the time comparisons SHOULD be performed in a manner consistent with the information in Section 7.4.

- If both the existing and the new record contain client-last-transaction-time information, the record with the later client-last-transaction-time is considered better.

- If one of the records contains client-last-transaction-time information and the other one doesn't, then compare the client-last-transaction-time in the record that contains it against the other record's start-time-of-state. The record with the later time is considered better.

- If neither record contains client-last-transaction-time information, compare their start-time-of-state information. The record with the later start-time-of-state is considered better.

- If none of the comparisons above yield a clear answer as to which record is later, then compare the value of the REMOTE flag from the data-source option for each record. If the values of the REMOTE flag are different between the two records, the record with the REMOTE flag value of local is considered better.

The above algorithm does not necessarily determine which record is better. In the event that the algorithm is inconclusive with regard to a record that was just received by the requestor, the requestor SHOULD use additional information in the two records to make a determination as to which record is better.

7.7. Multiple Queries to a Single Server over One Connection

Bulk Leasequery requestors may need to make multiple queries in order to recover binding information. A requestor MAY use a single connection to issue multiple queries to a server willing to support them. Each query MUST have a unique xid.

A server SHOULD allow configuration of the number of queries that can be processed simultaneously over a single connection. A server SHOULD read the number of queries it is configured to process simultaneously and only read any subsequent queries as current queries are processed.

A server that is processing multiple queries simultaneously MUST NOT block sending replies on new queries until all replies for the existing query are complete. Requestors need to be aware that replies for multiple queries may be interleaved within the stream of reply messages. Requestors that are not able to process interleaved replies (based on xid) MUST NOT send more than one query over a single connection prior to the completion of the previous query.

Requestors should be aware that servers are not required to process more than one query over a connection at a time (the limiting case for the configuration described above) and that servers are likely to limit the rate at which they process queries from any one requestor.

7.7.1. Example

This example illustrates what a series of queries and responses might look like. This is only an example -- there is no requirement that this sequence must be followed or that requestors or servers must support parallel queries.

In the example session, the client sends four queries after establishing a connection. Query 1 returns no results; query 2 returns 3 messages, and the stream of replies concludes before the client issues any new query. Query 3 and query 4 overlap, and the server interleaves its replies to those two queries.

```
  Requestor                              Server
  ---------                              ------
  DHCPBULKLEASEQUERY xid 1 ----->
                           <-----        DHCPLEASEQUERYDONE xid 1
  DHCPBULKLEASEQUERY xid 2 ----->
                           <-----        DHCPLEASEACTIVE xid 2
                           <-----        DHCPLEASEACTIVE xid 2
                           <-----        DHCPLEASEACTIVE xid 2
                           <-----        DHCPLEASEQUERYDONE xid 2
  DHCPBULKLEASEQUERY xid 3 ----->
  DHCPBULKLEASEQUERY xid 4 ----->
                           <-----        DHCPLEASEACTIVE xid 4
                           <-----        DHCPLEASEACTIVE xid 4
                           <-----        DHCPLEASEACTIVE xid 3
                           <-----        DHCPLEASEACTIVE xid 4
                           <-----        DHCPLEASEUNASSIGNED xid 3
                           <-----        DHCPLEASEACTIVE xid 4
                           <-----        DHCPLEASEACTIVE xid 3
                           <-----        DHCPLEASEQUERYDONE xid 3
                           <-----        DHCPLEASEACTIVE xid 4
                           <-----        DHCPLEASEQUERYDONE xid 4
```

7.8. Closing Connections

If a requestor has no additional queries to send, or doesn’t know if it has additional queries to send or not, then it SHOULD close the connection after receiving the DHCPLEASEQUERYDONE message for the last outstanding query that it sent.

The requestor SHOULD close connections in a graceful manner and not an abort. The requestor SHOULD NOT assume that the manner in which the DHCP server closed a connection carries any special meaning.

Typically, the requestor is the entity that will close the connection, as servers will often wait with an open connection in case the requestor has additional queries.

If a server closes a connection with an exception condition, the requestor SHOULD consider as valid any completely received intermediate results, and the requestor MAY retry the Bulk Leasequery operation.

8. Server Behavior

8.1. Accepting Connections

Servers that implement DHCPv4 Bulk Leasequery listen for incoming TCP connections. Port numbers are discussed in Section 6.3. Servers MUST be able to limit the number of concurrently accepted and active connections. The value BULK_LQ_MAX_CONNS SHOULD 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 Bulk Leasequery connections and DHCPBULKLEASEQUERY messages to certain requestors. Connections not from permitted requestors SHOULD be closed immediately to avoid server connection resource exhaustion. Servers MAY restrict some requestors to certain query types. Servers MAY reply to queries that are not permitted with the DHCPLEASEQUERYDONE message with a status-code option status of NotAllowed or MAY simply close the connection.

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 a Bulk 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 a BULK_LQ_DATA_TIMEOUT.

Every Bulk Leasequery request MUST be terminated by sending a final DHCPLEASEQUERYDONE message if such a message can be sent. The DHCPLEASEQUERYDONE message MUST have a status-code option status if the termination was other than successful, and SHOULD NOT contain a status-code option status if the termination was successful.

If the DHCPv4 server encounters an error during processing of the DHCPBULKLEASEQUERY message, either during initial processing or later during the message processing, it SHOULD send a DHCPLEASEQUERYDONE containing a status-code option. It MAY close the connection after this error is signaled, but that is not required.

If the server does not find any bindings satisfying a query, it MUST send a DHCPLEASEQUERYDONE. It SHOULD NOT include a status-code option with a Success status unless there is a useful string to include in the status-code option. Otherwise, the server sends each binding’s data in a DHCPLEASEACTIVE or DHCPLEASEUNASSIGNED message.

The response to a DHCPBULKLEASEQUERY may involve examination of multiple DHCPv4 IP address bindings maintained by the DHCPv4 server. The Bulk Leasequery protocol does not require any ordering of the IP addresses returned in DHCPLEASEACTIVE or DHCPLEASEUNASSIGNED messages.

When responding to a DHCPBULKLEASEQUERY message, the DHCPv4 server MUST NOT send more than one message for each applicable IP address, even if the state of some of those IP addresses changes during the processing of the message. Updates to such IP address state are already handled by normal protocol processing, so no special effort is needed here.

If the ciaddr, yiaddr, or siaddr is non-zero in a DHCPBULKLEASEQUERY request, the request must be terminated immediately by a DHCPLEASEQUERYDONE message with a status-code option status of MalformedQuery.

Any DHCPBULKLEASEQUERY that has more than one of the following primary query types specified MUST be terminated immediately by a DHCPLEASEQUERYDONE message with a status-code option status code of NotAllowed.

The allowable queries in a DHCPBULKLEASEQUERY message are processed as follows. Note that the descriptions of the primary queries below must be constrained by the actions of any of the three qualifiers described subsequently as well.

The following table discusses how to process the various queries. For information on how to identify the query, see the information in Section 7.2.

- Query by MAC address

  Every IP address that has a current binding to a DHCPv4 client matching the chaddr, htype, and hlen in the DHCPBULKLEASEQUERY request MUST be returned in a DHCPLEASEACTIVE message.

- Query by Client-identifier

  Every IP address that has a current binding to a DHCPv4 client matching the Client-identifier option in the DHCPBULKLEASEQUERY request MUST be returned in a DHCPLEASEACTIVE message.

- Query by Remote ID

  Every IP address that has a current binding to a DHCPv4 client matching the Remote ID sub-option of the relay-agent-information option in the DHCPBULKLEASEQUERY request MUST be returned in a DHCPLEASEACTIVE message.

- Query by Relay-ID

  Every IP address that has a current binding to a DHCPv4 client matching the Relay-ID sub-option of the relay-agent-information option in the DHCPBULKLEASEQUERY request MUST be returned in a DHCPLEASEACTIVE message.

- Query for All Configured IP Addresses

  A Query for All Configured IP addresses is signaled by the absence of any other primary query. That is, if there is no value in the chaddr, hlen, htype, no Client-identifier option, and no Remote ID sub-option or Relay-ID sub-option of the relay-agent-information option, then the request is a query for information concerning all configured IP addresses. In this case, every configured IP address that has a current binding to a DHCPv4 client MUST be returned in a DHCPLEASEACTIVE message. In addition, every configured IP address that does not have a current binding to a DHCPv4 client MUST be returned in a DHCPLEASEUNASSIGNED message.

  In this form of query, each configured IP address MUST be returned at most one time. In the absence of qualifiers restricting the number of IP addresses returned, every configured IP address MUST be returned exactly once.

There are three qualifiers that can be applied to any of the above primary queries. These qualifiers can appear individually or together in any combination, but only one of each can appear.

- Query Start Time

  If a query-start-time option appears in the DHCPBULKLEASEQUERY request, only IP address bindings that have changed on or after the time specified in the query-start-time option should be returned.

- Query End Time

  If a query-end-time option appears in the DHCPBULKLEASEQUERY request, only IP address bindings that have changed on or before the time specified in the query-end-time option should be returned.

- VPN-ID

  If no VPN-ID option appears in the DHCPBULKLEASEQUERY, the default (global) VPN is used to satisfy the query. A VPN-ID option [RFC6607] value other than the wildcard value (254) allows the requestor to specify a single VPN other than the default VPN. In addition, the VPN-ID option has been extended as part of this document to allow specification of a type 254, which indicates that all configured VPNs be searched in order to satisfy the primary query.

  In all cases, if the information returned in a DHCPLEASEACTIVE or DHCPLEASEUNASSIGNED message is for a VPN other than the default (global) VPN, a VPN-ID option MUST appear in the packet.

The query-start-time and query-end-time qualifiers are used to constrain the amount of data returned by a Bulk Leasequery request by returning only IP addresses whose address bindings have changed in some way during the time window specified by the query-start-time and query-end-time.

A DHCPv4 server SHOULD consider an address binding to have changed during a specified time window if either the client-last-transaction-time or the start-time-of-state of the address binding changed during that time window.

The DHCPv4 server MAY return address binding data in any order, as long as binding information for any given IP address is not repeated. When all binding data for a given DHCPBULKLEASEQUERY has been sent, the DHCPv4 server MUST send a DHCPBULKLEASEQUERYDONE message.

8.3. Building a Single Reply for Bulk Leasequery

The DHCPv4 Leasequery specification [RFC4388] describes the initial construction of DHCPLEASEQUERY reply messages using the DHCPLEASEACTIVE and DHCPLEASEUNASSIGNED message types in Section 6.4.2. All of the reply messages in Bulk Leasequery are similar to the reply messages for an IP address query. Message transmission and framing for TCP are described in this document in Section 6.1.

[RFC2131] and [RFC4388] specify that every response message MUST contain the server-identifier option. However, that option will be the same for every response from a particular DHCPBULKLEASEQUERY request. Thus, the DHCPv4 server MUST include the server-identifier option in the first message sent in response to a DHCPBULKLEASEQUERY. It SHOULD NOT include the server-identifier option in later messages.

The message type of DHCPLEASEACTIVE or DHCPLEASEUNASSIGNED is based on the value of the dhcp-state option. If the dhcp-state option value is ACTIVE, then the message type is DHCPLEASEACTIVE; otherwise, the message type is DHCPLEASEUNASSIGNED.

In addition to the basic message construction described in [RFC4388], the following guidelines exist:

1. If the dhcp-state option code appears in the dhcp-parameter-request-list, the DHCPv4 server SHOULD include a dhcp-state option whose value corresponds most closely to the state held by the DHCPv4 server for the IP address associated with this reply. If the state is ACTIVE and the message being returned is DHCPLEASEACTIVE, then the DHCPv4 server MAY choose to not send the dhcp-state option. The requestor SHOULD assume that any DHCPLEASEACTIVE message arriving without a requested dhcp-state option has a dhcp-state of ACTIVE.

2. If the base-time option code appears in the dhcp-parameter-request-list, the DHCPv4 server MUST include a base-time option, which is the current time in the DHCPv4 server’s context and the time from which the start-time-of-state, dhcp-lease-time, client-last-transaction-time, and other duration-style times are based upon.

3. If the start-time-of-state option code appears in the dhcp-parameter-request-list, the DHCPv4 server MUST include a start-time-of-state option whose value represents the time at which the dhcp-state option’s state became valid.

4. If the dhcp-lease-time option code appears in the dhcp-parameter-request-list, the DHCPv4 server MUST include a dhcp-lease-time option for any state that has a timeout value associated with it.

5. If the data-source option code appears in the dhcp-parameter-request-list, the DHCPv4 server MUST include the data-source option in any situation where any of the bits would be non-zero. Thus, in the absence of the data-source option, the assumption is that all of the flags are zero.

6. If the client-last-transaction-time option code appears in the dhcp-parameter-request-list, the DHCPv4 server MUST include the client-last-transaction-time option in any situation where the information is available.

7. If there is a dhcp-parameter-request-list in the initial DHCPBULKLEASEQUERY request, then it should be used for all of the replies generated by that request. Some options can be sent from a DHCPv4 client to the server or from the DHCPv4 server to a DHCPv4 client. Option 125 is such an option. If the option code for one of these options appears in the dhcp-parameter-request-list, it SHOULD result in returning the value of the option sent by the DHCPv4 client to the server if one exists.

Note that there may be other requirements for a reply to a DHCPBULKLEASEQUERY request, as discussed in Section 8.2.

8.4. Multiple or Parallel Queries

As discussed in Section 7.3, requestors may want to use a connection that has already been established when they need to make additional queries. Servers SHOULD support reading and processing multiple queries from a single connection and SHOULD allow configuration of the number of simultaneous queries it may process. A server MUST NOT read more query messages from a connection than it is prepared to process simultaneously.

This SHOULD be a feature that is administratively controlled. Servers 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.

8.5. Closing Connections

The DHCPv4 server SHOULD close connections in a graceful manner and not abort the connection. The DHCPv4 server SHOULD NOT assume that the manner in which the requestor closed a connection carries any special meaning.

Typically, the DHCPv4 server will only close the connection after some form of an exception or a timeout on the connection.

Using a timer to detect when a connection is idle and then closing that connection is designed to protect the DHCPv4 server from consuming unnecessary resources.

The DHCPv4 server should start a timer for BULK_LQ_DATA_TIMEOUT seconds for a particular connection after it sends a DHCPLEASEQUERYDONE message over that connection if there is no current query outstanding for that connection. It should restart this timer if a query arrives over that connection. If the timer expires, the DHCPv4 server should close the connection.

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 status-code option in a DHCPLEASEQUERYDONE message, including a message string indicating details of the reason for the abort. If the connection is closed for any reason, all of the data flows associated with any currently outstanding DHCPBULKLEASEQUERY messages will be terminated.

If the server detects that the requesting end of the connection 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 authentication of LEASEQUERY messages, mitigation of packet-flooding DoS attacks, and restriction to trusted requestors.

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 requestors to receive service. Malicious requestors who succeed in establishing connections but who then send invalid

queries, partial queries, or no queries at all can also exhaust a server's pool of available connections. We recommend that servers offer configuration to limit the sources of incoming connections, that they limit the number of accepted connections and the number of in-process queries from any one connection, and that they limit the period of time during which an idle connection will be left open.

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

To prevent information leakage to unauthorized clients, servers SHOULD restrict Bulk Leasequery connections and DHCPBULKLEASEQUERY messages to certain requestors, either through explicit configuration of the server itself or by employing external network elements to provide such restrictions. In particular, the typical DHCPv4 client SHOULD NOT be allowed to receive a response to a Bulk Leasequery request, and some technique MUST exist to allow prevention of such access in any environment where Bulk Leasequery is deployed.

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

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

Authentication for DHCP messages [RFC3118] MUST NOT be used to attempt to secure transmission of the messages described in this document. In particular, the message framing would not be protected by using the mechanisms described in [RFC3118] (which was designed only with UDP transport in mind).

10. IANA Considerations

IANA has assigned the following new DHCPv4 option codes from the registry "BOOTP Vendor Extensions and DHCP Options" maintained at http://www.iana.org/assignments/bootp-dhcp-parameters.

1. An option code of 151 for status-code.

2. An option code of 152 for base-time.

3. An option code of 153 for start-time-of-state.

4. An option code of 154 for query-start-time.

5. An option code of 155 for query-end-time.

6. An option code of 156 for dhcp-state.

7. An option code of 157 for data-source.

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 14 for DHCPBULKLEASEQUERY.

2. A dhcp-message-type of 15 for DHCPLEASEQUERYDONE.

IANA has created a new registry on the same assignments page, titled "DHCP State 156 Values" (where 156 corresponds to the assigned value of the dhcp-state option above). This registry has the following initial values:

State	
1	AVAILABLE
2	ACTIVE
3	EXPIRED
4	RELEASED
5	ABANDONED
6	RESET
7	REMOTE
8	TRANSITIONING

New values for this namespace may only be defined by IETF Review, as described in [RFC5226].

IANA has created a new registry on the same assignments page, titled "DHCP Status Code 151 Values" (where 151 corresponds to the assigned value of the status-code option above). This registry has the following initial values:

Name	status-code
Success	000
UnspecFail	001
QueryTerminated	002
MalformedQuery	003
NotAllowed	004

New values for this namespace may only be defined by IETF Review, as described in [RFC5226].

IANA has revised the registry "VSS Type Options" created by [RFC6607] in the overall area "Dynamic Host Configuration Protocol (DHCP) and Bootstrap Protocol (BOOTP) Parameters". It has been revised to appear as follows. Note that the number range for "Unassigned" has changed, and a new line for "All VPNs (wildcard)" was added.

Type	VSS Information Format
0	Network Virtual Terminal (NVT) ASCII VPN identifier
1	RFC 2685 VPN-ID
2-253	Unassigned
254	All VPNs (wildcard)
255	Global, default VPN

11. Acknowledgements

Significant text as well as important ideas were borrowed in whole or in part from "DHCPv6 Bulk Leasequery" [RFC5460], written by Mark Stapp. Further suggestions and improvements were made by participants in the DHC Working Group, including Alfred Hoenes.

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.

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

[RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.

[RFC3118] Droms, R., Ed., and W. Arbaugh, Ed., "Authentication for DHCP Messages", RFC 3118, June 2001.

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5735] Cotton, M. and L. Vegoda, "Special Use IPv4 Addresses", BCP 153, RFC 5735, January 2010.

[RFC6607] Kinnear, K., Johnson, R., and M. Stapp, "Virtual Subnet Selection Options for DHCPv4 and DHCPv6", RFC 6607, April 2012.

[RFC6925] Joshi, B., Desetti, R., and M. Stapp, "The DHCPv4 Relay Agent Identifier Sub-Option", RFC 6925, April 2013.

12.2. Informative References

[RFC951] Croft, W. and J. Gilmore, "Bootstrap Protocol", RFC 951, September 1985.

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

[RFC4614] Duke, M., Braden, R., Eddy, W., and E. Blanton, "A Roadmap for Transmission Control Protocol (TCP) Specification Documents", RFC 4614, September 2006.

[RFC5460] Stapp, M., "DHCPv6 Bulk Leasequery", RFC 5460, February 2009.

Authors' Addresses

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

Phone: (978) 936-0000
EMail: kkinnear@cisco.com

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

Phone: (978) 936-0000
EMail: mjs@cisco.com

D.T.V Ramakrishna Rao
Infosys Ltd.
44 Electronics City, Hosur Road
Bangalore 560 100
India

EMail: ramakrishnadtv@infosys.com
URI: http://www.infosys.com/

Bharat Joshi
Infosys Ltd.
44 Electronics City, Hosur Road
Bangalore 560 100
India

EMail: bharat_joshi@infosys.com
URI: http://www.infosys.com/

Neil Russell
Sea Street Technologies Inc.

EMail: neil.e.russell@gmail.com

Pavan Kurapati
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA

EMail: kurapati@juniper.net
URI: http://www.juniper.net/

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

Phone: (978) 936-0000
EMail: volz@cisco.com