DHC Working Group
Internet Draft
Intended Status: Standards Track
Expires: May 3, 2009

Kim Kinnear
Bernie Volz
Neil Russell
Mark Stapp
Cisco Systems, Inc.
D. Rao
B. Joshi
P. Kurapati
Infosys Technologies Ltd.
November 3, 2008

Bulk DHCPv4 Lease Query
<draft-kinnear-dhc-dhcpv4-bulk-leasequery-01.txt>

Abstract

The Dynamic Host Configuration Protocol for IPv4 (DHCPv4) has been extended with a Leasequery capability that allows a requestor to

request information about DHCPv4 bindings. That mechanism is limited to queries for individual bindings. In some situations individual binding queries may not be efficient, or even possible. This document expands on the DHCPv4 Leasequery protocol to allow for bulk transfer of DHCPv4 address binding data via TCP.

Table of Contents

1. Introduction

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

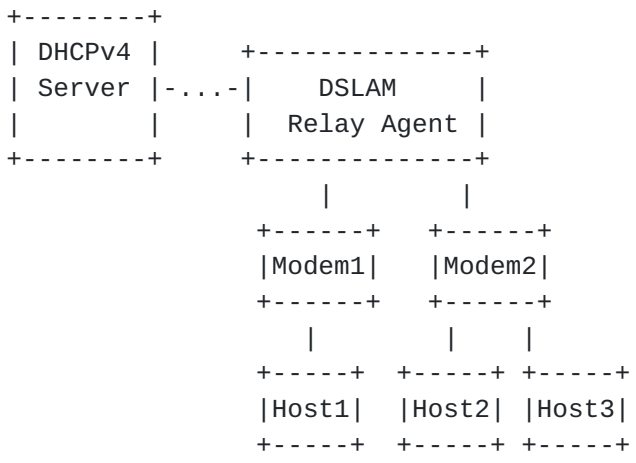


Figure 1: Example DHCPv4 configuration

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 the lease information sent by the server and maintain this locally. This information is used for a variety of purposes, including prevention of spoofing attempts from the DHCPv4 clients and to install 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 recover lease state information about a particular IP address or client in near real-time.

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.

Different query types are needed where a relay agent can query the server without waiting for the traffic from or for the clients, as well as a different transmission technique more conducive to the transmission of large quantities of data.

This document extends the DHCPv4 Leasequery protocol 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 could 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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This document uses the following terms:

- "absolute time"

  A 32-bit quantity containing the number of seconds since Jan 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.

- "active binding"

  An IP address with an active binding refers to an IP address which is currently associated with a DHCPv4 client where that

DHCPv4 client has the right to use the IP address.

- "Bulk Leasequery"

  Requesting and receiving the existing DHCPv4 address binding information in an efficient manner.

- "clock skew"

  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 is termed the "clock skew" for that Bulk Leasequery connection. It is not absolutely constant but is likely to vary only slowly. It is possible that, when both systems run NTP, that the clock skew is zero, and 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.

- "DHCPv4 client"

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

- "DHCPv4 relay agent"

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

- "DHCPv4 server"

  A DHCPv4 server is an Internet host that returns configuration parameters to DHCPv4 clients.

- "downstream"

  Refers to a direction away from the central part of a network and toward the edge. In a DHCPv4 context, typically refers to a network direction which is away from the DHCPv4 server.

- "IP address"

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

- "IP address binding"

  The information that a DHCPv4 server keeps regarding the relationship between a DHCPv4 client and an IPv4 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 IPv4 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 MAC address consists of the fields: hardware type "htype", hardware length "hlen", and client hardware address "chaddr".

- "upstream"

  Refers to a direction toward the central part of a network and away from the edge. In a DHCPv4 context, typically refers to a network direction which is 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 which 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. Motivation

Consider a typical DSLAM working also as a DHCPv4 relay agent (see Figure 1). Typically, both a "fast path" and a "slow path" exist in many network elements, including DSLAMs. Fast path processing is done in a network processor or in an ASIC (Application Specific

Integrated Circuit). Slow path processing is done in a normal processor. As much as possible, regular data handling code should be in the fast path. Slow path processing should be reduced as it may become a bottleneck.

For a DSLAM having multiple DSL ports, multiple IP addresses may be assigned using DHCPv4 to a single port and the number of DHCPv4 clients on a port may be unknown. The DSLAM may also not know the network portions of the IP addresses that are assigned to its DHCPv4 clients.

The DSLAM gleans IP address or other information from DHCP negotiations for antispoofing and for other purposes. The antispoofing itself is done in the fast path. The DSLAM keeps track of only one list of IP addresses: the list of IP addresses that are assigned by a DHCPv4 server. Traffic for all other IP addresses is dropped. If a client starts its data transfer after its DHCPv4 negotiations are gleaned by the DSLAM, no legitimate packets will be dropped because of antispoofing. In other words, antispoofing is effective (no legitimate packets are dropped and all spoofed packets are dropped) and efficient (antispoofing is done in the fast path). The intention is to achieve similar effective and efficient antispoofing in the Leasequery scenario after a DSLAM loses its gleaned information (for example, because of reboot).

After a deep analysis, we found that the three existing query types supported by [RFC4388] do not provide effective and efficient antispoofing for the above scenario and a new mechanism is required.

The existing query types

- necessitate a data driven approach: the lease queries can only be done when the Access Concentrator receives data. That results in increased outage time for DHCPv4 clients.

- result in excessive negative caching consuming lot of resources under a spoofing attack.

- result in antispoofing being done in the slow path instead of the fast path.

- do not support an Access Concentrator which periodically uploads its internal table to some form of stable storage

The deeper analysis, which led to the above conclusions, itself appears as an Appendix to this document.

4. Design Goals

The goal of this document is to provide a lightweight mechanism for an Access Concentrator or other network element to retrieve IP address binding information available in the DHCPv4 server. The mechanism should also allow an Access Concentrator to retrieve consolidated IP address binding information for the entire access concentrator or for a single connection/circuit.

4.1. 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, packets need to be dropped until it gets the lease information from DHCPv4 server. If an Access Concentrator finishes the Leasequeries before it starts receiving data, then there is no need to drop legitimate packets. In this way, outage time may be reduced.

4.2. Lessen Negative Caching

If Leasequeries result in negative caches, then that puts additional overhead on the access concentrator. The negative caches not only consume precious resources, they also need to be managed. Hence they should be avoided as much as possible. The Leasequeries should reduce the need for negative caching as far as possible.

4.3. Antispoofing in 'Fast Path'

If Antispoofing is not done in 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 fast path.

4.4. 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 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.

5. Protocol Overview

The Bulk Leasequery mechanism is modeled on the existing individual Leasequery protocol in [RFC4388] as well as related work on DHCPv6 Bulk Leasequery [DHCPv6Bulk]. 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.

The Bulk Leasequery mechanism 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 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 DUID carried in a Relay-ID sub-option [RelayId]. 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). In the event these times 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 [VpnId] 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. By default, only the default VPN is used to satisfy the query.

6. 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.

Only the DHCPBULKLEASEQUERY request is supported over the Bulk Leasequery connection. No other DHCPv4 requests are supported. The Bulk Leasequery connection is not an alternative DHCPv4 communication option for clients seeking other DHCPv4 services.

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-id.

One change in behavior for these existing queries is required when Bulk Leasequery is used. [RFC4388], in sections 6.1, 6.4.1, and 6.4.2 specifies 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.

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 and, as discussed above, the associated-ip option MUST NOT be used.

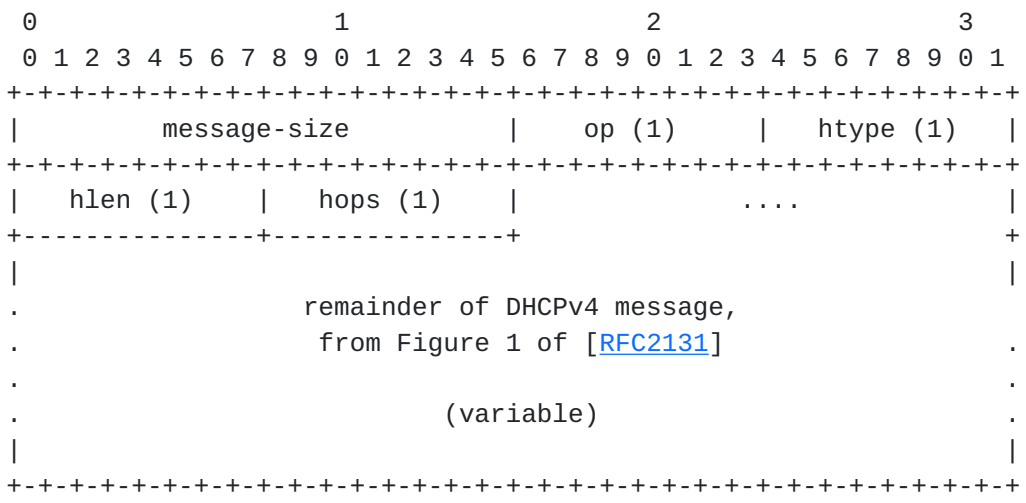
7. Message and Option Definitions

7.1. 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.

DHCPv4 message framed for TCP:



message-size the number of octets in the message that follows, as a 16-bit 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 which 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.

7.2. 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 which 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.

7.2.1. 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

7.2.2. dhcp-message

The dhcp-message option (option 56) from Section 9.9 of [RFC2132] requires additional definition for use in the context of a DHCPBULKLEASEQUERY.

The format of the NVT ASCII message in the dhcp-message option is specified to have the first three characters appear in a constrained format. The first three characters MUST be numeric (base 10) characters.

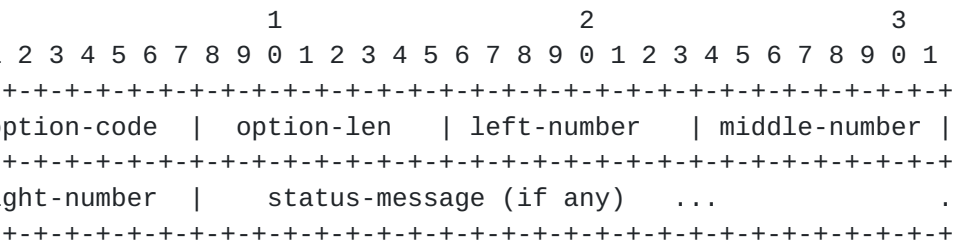
Encoded in these first three characters is the decimal number corresponding to a variety of status codes defined below.

The motivation for this constraint of the existing dhcp-message option is to reduce the number of top-level options used by this document.

The status code returned in the dhcp-message option allows greater detail to be returned regarding the status of a DHCPBULKLEASEQUERY request. While specified in the Bulk Leasequery document, this additional specification of the DHCPv4 dhcp-message option may well be valuable in other circumstances. In those circumstances its scope should be explicitly defined.

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 56. The length of this option is at least 3 octets.



option-code 56.

option-len 3 + length of status-message (which may be 0).

left-number, middle-number, right-number: NVT ASCII encoded characters representing the base-10 value of the status code, taken from the table below.

status-message: An optional NVT ASCII encoded text string suitable for display to an end user, which MUST NOT be null-terminated. It SHOULD start with an NVT ASCII space.

Name	status-code	Description
Success	000	Success. Also signaled by absence of dhcp-message 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.

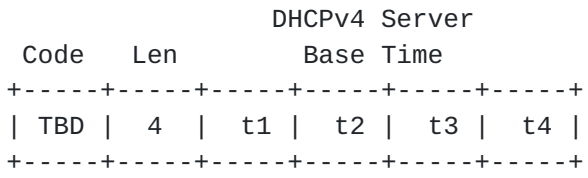
A dhcp-message option MAY appear in the options field of a DHCPv4 message. If the dhcp-message option does not appear, it is assumed that the operation was successful. The dhcp-message option SHOULD NOT appear in a message which is successful unless there is some text string that needs to be communicated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

7.2.3. 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.

This is an integer in network byte order.

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



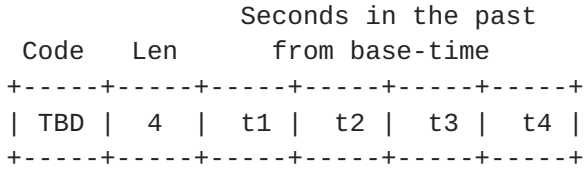
7.2.4. start-time-of-state

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

This MUST NOT be an absolute time. This MUST NOT be an absolute number of seconds since Jan 1, 1970. Instead, this MUST be an integer number of seconds in the past from the time specified in the base-time option in the same message that the IP address transitioned into its current state. In the same way that the IP Address Lease Time option (option 51) encodes a lease time which is a number of seconds into the future from the time the message was sent, this option encodes a value which is a number of seconds into the past from the base-time option included in the same message.

This is an integer in network byte order.

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



7.2.5. query-start-time

The query-start-time option allows the requestor to specify 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.

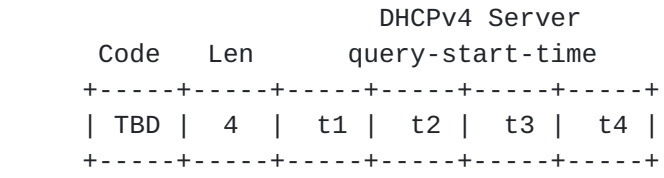
This MUST be an absolute time.

This MUST be a 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.

It SHOULD NOT be a time in the context of the requestor.

This is an integer in network byte order.

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



7.2.6. query-end-time

The query-end-time option allows the requestor to specify 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.

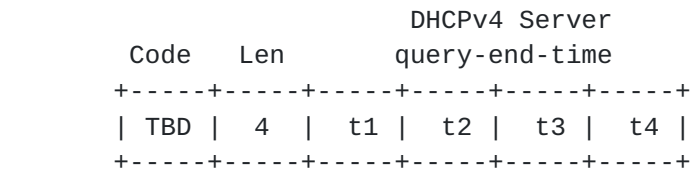
This MUST be an absolute time.

This MUST be a 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.

It SHOULD NOT be a time in the context of the requestor.

This is an integer in network byte order.

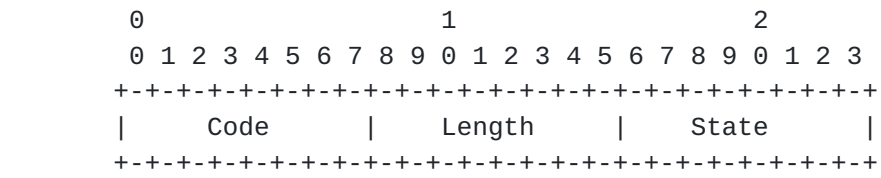
The code for this option is TBD. The length of this option is 4 octets.



7.2.7. 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 TBD. The length of this option is 1 octet.



Code The suboption code (TBD).

Length The suboption 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.

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 which 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.

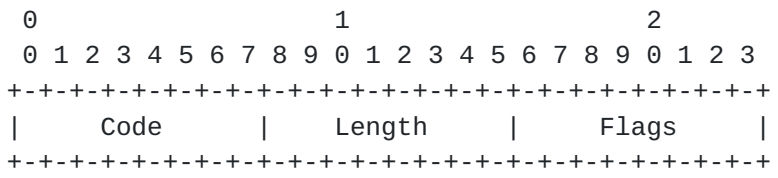
There is no requirement for the state of an IP address to transition in a well defined way from state to state. To put this another way, you cannot draw a simple state transition graph for the states of an IP address and the requestor of a Leasequery MUST NOT depend on one certain state always following a particular previous state. In general, every state can (at times) follow every other state.

7.2.8. data-source

The data-source option contains information about the source of the data in a DHCPLEASEACTIVE or a DHCPLEASEUNASSIGNED message. It is used when there are two or more servers who 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.

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.

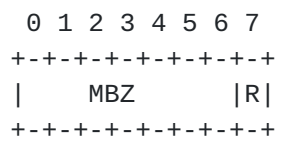
The code for this option is TBD. The length of this option is 1 octet.



Code The suboption code (TBD).

Length The suboption length, 1 octet.

Flags The Source information for this message.



R: REMOTE flag

remote = 1
local = 0

MBZ: MUST BE ZERO (reserved for future use)

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 the requestor SHOULD consider that the data-source was local.

7.2.9. Virtual Subnet Selection Type and Information

All of the (sub)options defined in [VpnId] 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 which indicates that all VPN's are to be used to process the Bulk Leasequery.

	Type	VSS Information format:
	0	NVT ASCII VPN identifier
	1	RFC2685 VPN-ID
	2-253	Not Allowed
NEW ->	254	All VPN's (wildcard).
	255	Global, default VPN.

7.3. 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.

Parameter	Default	Description
BULK_LQ_CONN_TIMEOUT	30 secs	Leasequery connection timeout
BULK_LQ_QUERY_TIMEOUT	30 secs	Leasequery query timeout
BULK_LQ_MAX_CONNS	10	Max Leasequery TCP connections
BULK_LQ_MAX_CONN_RETRY	60 secs	Max Leasequery retry timeout
BULK_LQ_DATA_TIMEOUT	30 secs	Leasequery data timeout

8. Requestor Behavior

8.1. Connecting and General Processing

A requestor attempts to establish a TCP connection to a DHCPv4 server in order to initiate a Leasequery exchange. The requestor SHOULD be prepared to abandon the connection attempt after BULK_LQ_CONN_TIMEOUT. If the attempt fails, the requestor MAY retry. Retries MUST use an exponential backoff timer, increasing the interval between attempts up to BULK_LQ_MAX_CONN_RETRY.

If Bulk Leasequery is terminated prematurely by a DHCPLEASEQUERYDONE with a dhcp-message status-code of QueryTerminated or by the failure of the connection over which it was being submitted, the requestor MAY retry the request after the creation of a new connection. Retries MUST use an exponential backoff timer, increasing the interval between attempts up to BULK_LQ_MAX_CONN_RETRY.

Messages from the DHCPv4 server come as multiple responses to a single DHCPBULKLEASEQUERY message. Thus, each DHCPBULKLEASEQUERY request MUST have a 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 different DHCPBULKLEASEQUERY requests to be demultiplexed by the requestor.

A requestor MAY send a DHCPBULKLEASEQUERY 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 Bulk Leasequery.

8.2. Forming a Bulk Leasequery

Bulk Leasequery is designed to create a connection which will transfer the state of some subset (or possibly all) of the IP address bindings to the requestor from DHCPv4 server. 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 which meet the specified query criteria. For some query criteria, this may include IP address binding information for IP addresses which may not now have or ever had have 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 of a series of primary queries and the optional addition of one or more qualifiers to those primary queries.

The possible primary queries are listed below. Each DHCPBULKLEASEQUERY request MUST consist of 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-Id

  In a Query by Client-Id, the dhcp-client-id option containing the requested value is included in the DHCPBULKLEASEQUERY request.

- Query by Remote-Id

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

- Query by Relay-Id

  In a Query by Relay-Id, the relay-id sub-option [RelayId] of the relay-agent-information option containing the requested value is included in 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 which 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 the query-start-time option specifies that only IP address bindings which have changed on or after the time specified in the query-start-time option should be returned.

- Query End Time

  Inclusion of the query-end-time option specifies that only IP address bindings which 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 VPN is used to search to satisfy the query specified by the DHCPBULKLEASEQUERY. Using the vpn-id option [VpnId] 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 VPN's 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 VPN MUST contain a vpn-id option in the message.

Both of the query-start-time and query-end-time options (if they appear) MUST be in the time context of the DHCPv4 server to which the Bulk Leasequery is directed. In the absence of information to the contrary, the requestor SHOULD assume that the time context on the DHCPv4 server is identical to the time context on the requestor. In the event that previous operations have determined that the time context on the DHCPv4 server to which the Bulk Leasequery is addressed differs from the time context of the requestor, the time context of the DHCPv4 server MUST be used.

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.

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

8.3. Processing Bulk Replies

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

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 a response message does not contain a DHCPv4 server-identifier option (option 54), then the server-identifier option from the previous message should be used. Thus, the DHCPv4 server MUST send the server-identifier option in the first response message, and MAY send it in subsequent response message for the same request.

The response messages generated by a DHCPBULKLEASEQUERY request are:

- o 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 dhcp-message option in the DHCPLEASEQUERYDONE message. If there is no dhcp-message option in the DHCPLEASEQUERYDONE message, then the query completed successfully.

  Note that a query which 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.

- o DHCPLEASEACTIVE

  A Bulk Leasequery will generate DHCPLEASEACTIVE messages containing binding data for bound IP addresses which match the specified query criteria. The IP address which 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, and hlen and possibly additional options.

- o DHCPLEASEUNASSIGNED

  Some queries will also generate DHCPLEASEUNASSIGNED messages for IP addresses which match the query criteria. These messages indicate that the IP address was 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, and hlen and possibly additional options.

- DHCPLEASEUNKNOWN

  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 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, and so the associated-ip option is not used.

8.4. 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, and 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 need to ever issue a DHCPBULKLEASEQUERY message using those options as part of the query.

In addition to saving the base-time for possible future use in a

query-start-time option, the base-time is used as part of the conversion of the other times in the Leasequery message to values which are meaningful in the context of the requestor.

The requestor SHOULD use the base-time values received in Bulk Leasequery messages to develop a value which represents the clock skew between the DHCPv4 server and the requestor. In theory this clock skew would simply be the difference between the first base-time value and the current time on the requestor when the message containing the base-time value was received. However, there may be transmission delays at the beginning or end or along the TCP connection, and so the actual clock skew may not be the same as any individual difference between a base-time value and the current time of the requestor.

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

The requestor SHOULD smooth the value which it uses as the clock skew by continuously examining the instantaneous value developed from the base-time of each message received from a DHCPv4 server and using this instantaneous value of clock skew to make small adjustments to the existing value of the clock skew. Thus, the clock skew will vary only slowly and one slow message will not completely distort a large number of future time calculations.

Given the value of the clock skew on the requestor, the requestor SHOULD bring all of the times in the DHCPLEASEACTIVE and DHCPLEASEUNASSIGNED messages into the context of the requestor. Except for the base-time value, the times in the Leasequery message are all relative to the base-time. These relative times SHOULD first be converted into absolute times in the context of the DHCPv4 server using the base-time value. Once this stage is complete, the absolute times that result SHOULD be brought into the context of the requestor by applying the calculated clock skew to each of the absolute times.

After all of this processing, the times are in the context of the requestor.

An alternative might appear to be to leave all of the times in the context of the DHCPv4 server, and if the requestor is dealing with only one DHCPv4 server at a time, this is an accurate and effective approach. However, if the requestor is dealing with DHCPLEASEACTIVE and DHCPLEASEUNASSIGNED messages from two or more different DHCPv4 servers, then in order to make any sense of them, the times from each server SHOULD be converted into the time of the requestor.

Since various transmission and processing delays may occur, a time converted into the requestor's context may be accurate to only a few seconds, at best. This is rarely an issue in the larger context of the use of the information derived from a Bulk Leasequery request. However, time comparison is an important factor in determining which update to the address binding information for a particular IPv4 address is the most recent and therefore worth remembering. The next section discusses the issue of comparing two updates in some detail, but a key aspect of that comparison is a comparison of the times in the two messages.

The requestor SHOULD consider times converted into its context as effectively equivalent if they are within a small number of seconds of each other. The precise number depends on the particular implementation involved, but 4 to 8 seconds is probably a good starting point. Thus, if two times are 3 seconds apart after conversion to the requestor's context they should be considered the same for purposes of comparison with each other.

8.5. 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 the Section 8.6 should be used.

8.6. 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 8.4.

- 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.

- o 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.

- o 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 which 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.

8.7. 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 MAY process more than one query at a time. A server that will not support more than one query at a time on a single connection MUST return a DHCPLEASEQUERYDONE message containing a dhcp-message option with a status-code of NotAllowed to the unsupported queries. Alternatively, a server that will not support more than one query at a time on a single connection MAY chose to simply read one query and only read any subsequent queries after processing of the current query is complete.

A server that is willing to do so MAY interleave replies to the multiple queries within the stream of reply messages it sends. 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, and that servers are likely to limit the rate at which they process queries from any one requestor.

8.7.1. 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
```

8.8. Closing Connections

Either the requestor or DHCPv4 server MAY close the TCP connection at any time. The requestor MAY choose to retain the connection if it intends to issue additional queries or if other queries are currently using the connection. Note that this requestor behavior does not guarantee that the connection will be available for additional queries: the server might decide to close the connection based on its own configuration.

9. Server Behavior

9.1. Accepting Connections

Servers that implement DHCPv4 Bulk Leasequery listen for incoming TCP connections. Port numbers are discussed in Section 7.3. Servers MUST be able to limit the number of currently 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 dhcp-message 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 an BULK_LQ_QUERY_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.

9.2. 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 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 dhcp-message status if the termination was other than successful, and SHOULD NOT contain a dhcp-message 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 dhcp-message 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 dhcp-message

option with a Success status unless there is a useful string to include in the dhcp-message 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.

A Bulk Leasequery response MUST contain no more than one message for each configured IP address in the DHCPv4 server. In addition, a Bulk Leasequery may well take significant time between the beginning and end of the processing of all of the messages required to satisfy the Bulk Leasequery query. During this time, the state of some of the IP addresses sent early in the response may change prior to the completion of the entire response to the Bulk Leasequery. This is normal and expected -- there is no requirement for the entire response to a Bulk Leasequery to represent an instantaneous snapshot of the state of the IP address bindings of a DHCPv4 server. Quite the contrary -- as the cursor moves through the IP addresses in whatever order is convenient to the DHCPv4 server, the state of IP addresses already examined can change and a DHCPv4 server MUST NOT try to examine IP addresses already scanned in an attempt to "keep up" with the ongoing state changes of all of the IP addresses. To do so would make it difficult to meet the requirement to send only one message per IP address in response to a Bulk Leasequery and would also make it difficult to know when to finish the Bulk Leasequery.

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 dhcp-message status of MalformedQuery.

Any DHCPBULKLEASEQUERY which has more than one of the following primary query types specified MUST be terminated immediately by a DHCPLEASEQUERYDONE message with a dhcp-message 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 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 8.2.

- Query by MAC address

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

- Query by Client-Id

  Every IP address which has a current binding to a DHCPv4 client which matches the client-id option in the DHCPBULKLEASEQUERY request MUST be returned in a DHCPLEASEACTIVE message.

- Query by Remote-Id

  Every IP address which has a current binding to a DHCPv4 client which matches 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 which has a current binding to a DHCPv4 client which matches 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-id option, 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 which has a current binding to a DHCPv4 client MUST be returned in a DHCPLEASEACTIVE message. In addition, every configured IP address which 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. If the absence of qualifiers which restrict the number of IP addresses returned, every configured IP address MUST be returned exactly once.

There are three qualifiers which 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.

o Query Start Time

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

o Query End Time

If a query-end-time option appears in the DHCPBULKLEASEQUERY request, only IP address bindings which 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 VPN is used to satisfy the query. A vpn-id option [VpnId] 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 VPN's be searched in order to satisfy the primary query.

In all cases, if the information returned in a DHCPLEASEACTIVE or DHCPLEASEUNASSIGNED message is for other than the default 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.

A DHCPv4 server MAY always compare the address binding information for an IP address against a time window if it follows the following guidelines. If there is no query-start-time, then the DHCPv4 server MUST assume the query-start-time is equivalent to a time prior to any time that resides in any IP address binding. If there is no query-end-time, the DHCPv4 server MUST assume that the query-end-time is equivalent to a time that is later than any time that resides in any IP address binding.

Even if the query-start-time or query-end-time option value is being used to limit the amount of data flow from the DHCPv4 server to the requestor, there is no requirement placed on the DHCPv4 server to return address binding data in any order and certainly not in any order based on time.

When the DHCPv4 server has no additional information to send to the requestor, it will send a DHCPLEASEQUERYDONE message.

9.3. Building a Single Reply for Bulk Leasequery

The DHCPv4 Leasequery [RFC4388] specification 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 is described in this document in Section 7.1.

[RFC2131] and [RFC4388] specify that every response message MUST contain the server-identifier option. However, that option will be the identical 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 MAY include the server-identifier in later messages as well, but there is no requirement for it to do so.

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 in 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 time-out value associated with it, and not just appear in a DHCPLEASEACTIVE message. Thus, the EXPIRED state which is sent in a DHCPLEASEUNASSIGNED message would have a dhcp-lease-time option in the message if the EXPIRED state represented a grace-period and would be changing state after the grace-period expired.

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 were 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 discussed in Section 9.2.

9.4. Multiple or Parallel Queries

As discussed in Section 8.3, requestors may want to leverage 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.

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

9.5. Closing Connections

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

The server MUST close its end of the TCP connection if it encounters an error sending data on the connection. The server MUST close its end of the TCP connection if it finds that it has to abort an in-process request. A server aborting an in-process request SHOULD attempt to signal that to its requestors by using the QueryTerminated status code in the dhcp-message option in a DHCPLEASEQUERYDONE message, including a message string indicating details of the reason for the abort. If the server detects that the requesting end of the connection has been closed, the server MUST close its end of the connection after it has finished processing any outstanding requests.

The server MUST send a DHCPLEASEQUERYDONE message at the end of the data returned from a Bulk Leasequery request.

10. Security Considerations

The "Security Considerations" section of [RFC2131] details the general threats to DHCPv4. The DHCPv4 Leasequery specification [RFC4388] describes recommendations for the Leasequery protocol, especially with regard to relayed LEASEQUERY messages, mitigation of packet-flooding DOS attacks, restriction to trusted requestors, and use of IPsec [RFC4301].

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

period of time during which an idle connection will be left open.

11. IANA Considerations

IANA is requested to assign the following new values for this document. See Section 7.2 for details.

1. A dhcp-message-type of 14 for DHCPBULKLEASEQUERY.

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

3. An option code of TBD for base-time.

4. An option code of TBD for start-time-of-state.

5. An option code of TBD for query-start-time.

6. An option code of TBD for query-end-time.

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

8. An option code of TBD for dhcp-state.

9. Values for dhcp-state:

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

10. Values for status code in a constrained dhcp-message option (option 53):

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

11. Addtional type field values for the Virtual Subnet Selection Type and Information [VpnId]:

	Type	VSS Information format:
	0	NVT ASCII VPN identifier
	1	RFC2685 VPN-ID
	2-253	Not Allowed
NEW ->	254	All VPN's. (wildcard)
	255	Global, default VPN.

12. Acknowledgements

This draft is a collaboration between the authors of draft-dtv-dhc-dhcpv4-bulk-leasequery-00.txt and draft-kkinnear-dhc-dhcpv4-bulk-leasequery-00.txt. Both documents acknowledged that significant text as well as ideas were borrowed in whole or in part from the DHCPv6 Bulk Leasequery draft [DHCPv6Bulk].

13. References

13.1. Normative References

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

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

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

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

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

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

[RelayId] Stapp, M., "The DHCPv4 Relay Agent Identifier Suboption", draft-ietf-dhc-relay-id-suboption-04.txt, September 2008.

[VpnId] Kinnear, K., R. Johnson, M. Stapp and J. Kumarasamy, "Virtual Subnet Selection Options for DHCPv4 and DHCPv6" draft-ietf-dhc-vpn-option-09.txt, July 2008.

13.2. Informative References

[RFC951] Croft, B., Gilmore, J., "Bootstrap Protocol (BOOTP)", RFC 951, September 1985.

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

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

[DHCPv6Bulk] Stapp, M., "DHCPv6 Bulk Leasequery", draft-ietf-dhc-dhcpv6-bulk-leasequery-04.txt, October 2008.

14. Authors' Addresses

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

Phone: (978) 936-0000

EMail: kkinnear@cisco.com

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

Phone: (978) 936-0000

EMail: volz@cisco.com

Neil Russell
Cisco Systems
1414 Massachusetts Ave.

Boxborough, Massachusetts 01719

Phone: (978) 936-0000

EMail: nrussell@cisco.com

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

Phone: (978) 936-0000

EMail: mjs@cisco.com

Ramakrishna Rao DTV
Infosys Technologies Ltd.
44 Electronics City, Hosur Road
Bangalore 560 100
India

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

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

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

Pavan Kurapati
Infosys Technologies Ltd.
44 Electronics City, Hosur Road
Bangalore 560 100
India

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

15. Full Copyright Statement

16. Intellectual Property

17. Acknowledgment

Funding for the RFC Editor function is provided by the IETF

Administrative Support Activity (IASA).

18. Appendix -- Why a New Leasequery is Required

The three existing query types supported by [RFC4388] do not provide effective and efficient antispoofing for the scenario discussed in Section 3.

o Query by Client Identifier

Query by Client Identifier is not possible because the DSLAM would need to glean the client-identifier. This is not possible since if we are using a Leasequery, it is because the gleaned information was lost. On the other hand, we can query by client-identifier when client sends a DHCPv4 request, but then there may not be any need for Leasequery as such -- regular gleaning may be enough.

o Query by IP Address

[RFC4388] suggests that it is preferable to use Query by IP Address when getting downstream traffic.

Query by IP address is not very useful because because downstream traffic may not exist for the clients on a DSL port. (In most Internet applications, downstream traffic exists only when a client sends upstream traffic). In other words, the client will be denied service until it gets downstream traffic, which may never come.

Query by IP address may be used for upstream traffic. Then whenever an upstream packet comes whose IP address is unknown to the DSLAM, a lease query may be initiated. A related question is what to do with that upstream traffic itself until lease query response comes? If the traffic is dropped, we may be dropping legitimate traffic. If the traffic is forwarded, we may be forwarding spoofed packets. Once the lease response comes, subsequent traffic is handled depending on the response. If a DHCPLEASEACTIVE response comes, the DSLAM will accept the traffic. If a DHCPLEASEUNASSIGNED response comes, the DSLAM will drop the traffic corresponding to the IP address. If a DHCPLEASEUNKNOWN response comes the DSLAM may drop the traffic corresponding to the IP address but will have to periodically send the lease query for that IP address again (additional overhead). The process is triggered whenever an unknown IP address comes.

Note that the DSLAM needs to keep track of 4 lists of IP addresses: (1) List of IP addresses for which it got DHCPLEASEACTIVE responses; (2) List of IP addresses for which it got DHCPLEASEUNASSIGNED responses; (3) List of IP addresses for which it got DHCPLEASEUNKNOWN responses; (4) All other IP addresses.

This approach may be acceptable if only legitimate traffic is received. Consider the case when someone sends packets that uses spoofed IP addresses. In that case, lease response will be DHCPLEASEUNASSIGNED or DHCPLEASEUNKNOWN. [RFC4388] suggests usage of negative caching in this regard (which involves additional resources).

In a spoofing type of attack, negative caching information may grow considerably if attacker varies the source IP address. For each such new source IP address, traffic will come to slow path, a new lease query needs to be initiated, response will be processed, and negative caching needs to be done. That will mean using many resources for negative caching.

[RFC4388] suggests that if the DSLAM knows the network portion of the IP addresses that are assigned to its clients, then some amount of antispoofing can be done in fast path and some lease queries may be avoided. But as indicated before, that information may not always be available to DSLAMs.

Effectively, antispoofing support involves considerable slow path processing and considerable resources tied for negative caching.

[RFC4388] says that DHCPv4 server should be protected from being flooded with too many Leasequery requests and DSLAM also should not send too many lease query messages at a time. This would mean that legitimate requestors may be excessively delayed getting their information in the face of antispoofing attacks.

It is concluded that antispoofing is neither effective nor efficient with this query type.

o Query by MAC Address

Query by MAC address can also be used in a way similar to query by IP address described above. Indeed, query by MAC address may be better than query by IP address in one sense because of the possible presence of the associated-ip option in lease responses. (Note that associated-ip option does not appear in responses for query by IP address). With associated-ip option DSLAM can get information not only about the IP address/MAC address that triggered the Leasequery but also about other IP addresses that are associated with the original MAC address. That way, when traffic that uses the other IP addresses comes along, DSLAM is already prepared to deal with them.

Although, query by MAC address is better than query by IP address in the above respect, it has a specific problem which is not shared by query by IP address. For a query by MAC address, only two types of

responses are possible: DHCPLEASEUNKNOWN and DHCPLEASEACTIVE; DHCPLEASEUNASSIGNED is not supported. This is particularly troublesome when a DHCPv4 server indeed has definitive information that no IP addresses are associated with the specified MAC address in the Leasequery, but it is forced to respond with DHCPLEASEUNKNOWN instead of DHCPLEASEUNASSIGNED. As we have seen above, unlike DHCPLEASEUNASSIGNED, DHCPLEASEUNKNOWN requires periodic querying with DHCPv4 server, an additional overhead.

Moreover, query by MAC address also shares all other issues we discussed above for query by IP address.

We conclude that existing Leasequery types are not appropriate to achieve effective and efficient antispoofing in the environment discussed.