DHC
Internet-Draft
Intended status: Standards Track
Expires: February 18, 2007

J. Brzozowski
Comcast Cable
K. Kinnear
B. Volz
S. Zeng
Cisco Systems, Inc.
August 17, 2006

DHCPv6 Leasequery
<draft-ietf-dhc-dhcvp6-leasequery-00.txt>

Abstract

This document specifies leasequery for the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) which can be used as a means to obtain lease information about DHCPv6 clients from a DHCPv6 server. This document specifies the scope of data that can be retrieved as well as both DHCPv6 leasequery requestor and server behavior. This document

extends DHCPv6.

Table of Contents

1. Introduction

The DHCPv6 [2] protocol specifies a mechanism for the assignment of both IPv6 address and configuration information to IPv6 nodes. IPv6 Prefix Options for DHCPv6 [4] specifies a mechanism for the automated delegation of IPv6 prefixes and related options. Similar to DHCPv4 [6], DHCPv6 servers maintain authoritative information related to its operations including but not limited to lease information for IPv6 addresses and delegated prefixes.

The requirement exists in various types of IPv6 deployments, particularly those of a broadband variety, to leverage DHCPv6 [2] for retrieving data related to the operation of DHCPv6 servers programmatically. In particular it is desirable to be able to extract lease information about IPv6 addresses and delegated prefixes assigned using DHCPv6 [2] [4]. Specific examples where this information has illustrated value are in broadband networks to facilitate access control by edge devices. This capability to programitcally extract lease data from the DHCPv6 server is called leasequery.

Existing specifications, such as [3] are leveraged as a basis for extending the DHCPv6 protocol to support leasequery. The motivations and justifications identified in [3] also generally apply to this specification. Furthermore, advancements in DHCPv6 [2] are expanded upon to specify additional means by which IPv6 address and delegated prefix lease data can be retrieved through DHCPv6 leasequery.

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 [1].

DHCPv6 terminology is defined in [2]. Terminology specific to DHCPv6 leasequery can be found below:

client(s) — The nodes that have one or more bindings with a DHCPv6 server. This does not refer to the node issuing the LEASEQUERY unless it itself has one or more bindings with a DHCPv6 server.

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

3. Protocol Overview

The focus of this document is to extend the DHCPv6 protocol to allow processes and devices that wish to access information from a DHCPv6 server to do so in a lightweight and convenient manner. It is especially appropriate for processes and devices that already interpret DHCPv6 messages.

The LEASEQUERY message is a query message only and does not affect the state of the IPv6 address or prefix, or the binding information associated with it.

One important motivating example is that the LEASEQUERY message allows access concentrators to query DHCP servers to obtain location information of broadband access network devices.

The leasequery capability described in this document parallels the DHCPv4 leasequery capability documented in [3]. As such, it shares many of the basic motivations, design goals and constraints as the capability described in Section 4 of [3].

3.1. On-Demand Query

The on-demand leasequery capability allows requesting just the information necessary to satisfy an immediate need. If the requestor is an access concentrator, then the immediate need will typically be that it has received an IPv6 packet and it needs to refresh its information concerning the DHCPv6 client to which that an IPv6 address is currently leased. In this case, the request will be by Address. This fits clearly into the single request/response cycle common to other DHCPv6 message exchanges.

However, this approach has limitations when used with prefix delegation [4] as no traffic may arrive because the access concentrator is unable to inject the appropriate routing information into the routing infrastructure, such as after a reboot. This approach does work if the access concentrator is configured to inject routing information for a prefix which aggregates potentially delegated prefixes. Or, if the access concentrator and requesting router use a routing protocol; as then the requesting router can trigger the access concentrator to request information from a DHCPv6 server and inject appropriate routing information into the routing infrastructure.

3.2. Anticipatory Query

A second approach for requesting information from a DHCPv6 server would be to use a leasequery-like capability to rebuild an internal

data store containing information available from a DHCPv6 server. The rebuilding of the data store in this approach can take place as soon as possible after the need to rebuild it is discovered (such as on booting), and doesn't wait on the receipt of specific packets to trigger a piecemeal database update (as is the case for on-demand leasequery). This approach would also remove the limitation discussed above for prefix delegation.

This anticipatory query is not specified in this document and is an area of future work.

4. Protocol Details

4.1. Message and Option Definitions

4.1.1. Messages

The LEASEQUERY and LEASEQUERY-REPLY messages use the Client/Server message formats described in [2], section 6. Two new message codes are defined:

LEASEQUERY (TBD) - A requestor sends a LEASEQUERY message to any available server to obtain information on a client's or clients' leases. The options in an OPTION_LQ_QUERY determine the query.

LEASEQUERY-REPLY (TBD) - A server sends a LEASEQUERY-REPLY message containing client data in response to a LEASEQUERY message.

4.1.2. Options

4.1.2.1. Query Option

The Leasequery Query option is used only in a LEASEQUERY message and identifies the query being performed. The option includes the query type, link-address (or 0::0 for no link), and option(s) to provide data needed for the query.

The format of the Query option is shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        OPTION_LQ_QUERY        |         option-len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   query-type  |                                               |
+-+-+-+-+-+-+-+-+                                               |
|                                                               |
|                         link-address                          |
|                                                               |
|               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               |                                               .
+-+-+-+-+-+-+-+-+                                               .
.                         query-options                         .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

option-code OPTION_LQ_QUERY (TBD)

option-len 17 + length of query-options field.

link-address A global address that will be used by the server to identify the link to which the query applies, or 0::0 if unspecified.

query-type the query requested (see below).

query-options the options related to the query.

The query-type and required query-options are:

QUERY_BY_ADDRESS (1) - The query-options MUST contain an OPTION_IAADDR option [2]. The link-address field, if not 0::0, MUST specify an address for the link on which the client is located if the address in the OPTION_IAADDR option is of insufficient scope. Only the information for the client that has a lease for the specified address or was delegated a prefix that contains the specified address is returned (if available).

The query-options MAY also include an OPTION_ORO option [2] to indicate the options for each client that the requestor would like the server to return. Note that this OPTION_ORO is distinct and separate from an OPTION_ORO that may be in the requestor's LEASEQUERY message.

If a server receives an OPTION_LQ_QUERY with a query-type it does not support, the server SHOULD return an UnknownQueryType status-code. If a server receives a supported query-type but the query-options is missing a required option, the server SHOULD return a MalformedQuery status-code.

4.1.2.2. Client Data Option

The Client Data option is used to encapsulate the data for a single client on a single link in a LEASEQUERY-REPLY message.

The format of the Client Data option is shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       OPTION_CLIENT_DATA      |         option-len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                       client-options                          .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
  option-code      OPTION_CLIENT_DATA (TBD)

  option-len       length, in octets, of the encapsulated client-
                   options field.

  client-options   the options associated with this client.
```

The encapsulated client-options include the OPTION_CLIENTID, OPTION_IAADDR, OPTION_IAPREFIX, and OPTION_CLT_TIME options and other options specific to the client and requested by the requestor in the OPTION_ORO in the OPTION_LQ_QUERY's query-options. The server MUST return all of the client's statefully assigned addresses and delegated prefixes, with a non-zero valid lifetime, on the link.

4.1.2.3. Client Last Transaction Time Option

The Client Last Transaction Time option is encapsulated in an OPTION_CLIENT_DATA and identifies how long ago the server last communicated with the client, in seconds.

The format of the Client Last Transaction Time option is shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        OPTION_CLT_TIME        |         option-len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 client-last-transaction-time                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

option-code   OPTION_CLT_TIME (TBD)

option-len   4

client-last-transaction-time
the number of seconds since the server last communicated with the client (on that link).

The client-last-transaction-time is a positive value and reflects the number of seconds since the server last communicated with the client (on that link).

4.1.3. Status Codes

The following new status codes are defined:

UnknownQueryType (TBD) - The query-type is unknown to or not supported by the server.

MalformedQuery (TBD) - The query is not valid, for example a required query-option is missing from the OPTION_LQ_QUERY.

NotConfigured (TBD) - The server does not have the target address or link in its configuration.

NotAllowed (TBD) - The server does not allow the requestor to issue this LEASEQUERY.

4.1.4. Transmission and Retransmission Parameters

This section presents a table of values used to describe the message transmission behavior for leasequery.

Parameter	Default	Description
LQ_TIMEOUT	1 sec	Initial LEASEQUERY timeout
LQ_MAX_RT	10 secs	Max LEASEQUERY timeout value
LQ_MAX_RC	5	Max LEASEQUERY retry attempts

4.2. Message Validation

4.2.1. LEASEQUERY

Requestors and clients MUST discard any received LEASEQUERY messages.

Servers MUST discard any received LEASEQUERY messages that meet any of the following conditions:

- o the message does not include an OPTION_CLIENTID option.
- o the message includes an OPTION_SERVERID option but the contents of the OPTION_SERVERID option does not match the server's identifier.
- o the message does not include an OPTION_LQ_QUERY option.

4.2.2. LEASEQUERY-REPLY

Requestors MUST discard any received LEASEQUERY-REPLY messages that meet any of the following conditions:

- o the message does not include an OPTION_SERVERID option.
- o the message does not include an OPTION_CLIENTID option or the contents of the OPTION_CLIENTID option do not match the DUID of the requestor.
- o the "transaction-id" field in the message does not match the value used in the original message.

Servers and Relay Agents (on the server port, 547 [2]) MUST discard any received LEASEQUERY-REPLY messages.

4.3. DHCPv6 Leasequery Requestor Behavior

This section describes how a requestor initiates lease data retrieval from DHCPv6 servers.

4.3.1. Creation of LEASEQUERY

The requestor sets the "msg-type" field to LEASEQUERY. The requestor generates a transaction ID and inserts this value in the "transaction-id" field.

The requestor MUST include an OPTION_CLIENTID option to identify itself to the server.

The requestor MUST include an OPTION_LQ_QUERY option and set the query-type, link-address, and query-options as appropriate to the query-type (Section 4.1.2.1).

The requestor SHOULD include an OPTION_SERVERID if it is not unicasting the LEASEQUERY yet only wants a response from a specific server.

4.3.2. Transmission of LEASEQUERY

The requestor MAY be configured to use a list of destination addresses, which MAY include unicast addresses, the All_DHCP_Servers multicast address, or other addresses selected by the network administrator. If the requestor has not been explicitly configured, it MAY use the All_DHCP_Servers multicast address as the default.

The requestor SHOULD send LEASEQUERY to one or more DHCPv6 servers which are known to possess authoritative information concerning the query target.

In the absence of information concerning which DHCPv6 servers might possess authoritative information on the query target, the requestor SHOULD send LEASEQUERY to all DHCPv6 servers that the requestor knows about or is configured with. For example, the requestor MAY send LEASEQUERY to the All_DHCP_Servers multicast address.

The requestor transmits LEASEQUERY messages according to section 14 of [2], using the following parameters:

```
IRT    LQ_TIMEOUT
MRT    LQ_MAX_RT
MRC    LQ_MAX_RC
MRD    0
```

If the message exchange fails, the requestor takes an action based on the requestor's local policy. Examples of actions the requestor might take include:

- o Select another server from a list of servers known to the requestor.
- o Send to multiple servers by multicasting to the All_DHCP_Servers address.
- o Terminate the leasequery.

4.3.3. Receipt of LEASEQUERY-REPLY

A successful LEASEQUERY-REPLY is one without an OPTION_STATUS_CODE with an error code and may or may not contain client data in OPTION_CLIENT_DATA options. A successful LEASEQUERY MAY contain no OPTION_CLIENT_DATA if no clients matched the query.

An unsuccessful LEASEQUERY-REPLY is one that has an OPTION_STATUS_CODE with an error code.

4.3.3.1. Receiving Successful LEASEQUERY-REPLY

Upon the receipt of a successful LEASEQUERY-REPLY in response to a LEASEQUERY, the requestor MUST extract the client data in the LEASEQUERY-REPLY and may update its binding information database.

The LEASEQUERY-REPLY SHOULD contain an OPTION_SERVER_RSN option [5] and the requestor SHOULD only update its binding information database as described in [5].

If an OPTION_CLIENT_DATA contains no OPTION_CLT_TIME, the requestor SHOULD silently discard the OPTION_CLIENT_DATA option.

The requestor MUST be prepared to handle an OPTION_CLIENT_DATA that contains more or fewer options than listed in the OPTION_ORO of the LEASEQUERY message.

4.3.3.2. Receiving Unsuccessful LEASEQUERY-REPLY

An unsuccessful LEASEQUERY-REPLY contains an OPTION_STATUS_CODE with one of the status codes listed in Section 4.1.3 or in [2] except Success.

Depending on the status code, the requestor may try a different server (such as for NotAllowed, NotConfigured, and UnknownQueryType) or try a different or corrected query (such as for UnknownQueryType and MalformedQuery).

4.3.4. Handling DHCPv6 Client Data from Multiple Sources

A requestor may receive lease data on the same client from the same DHCPv6 server in response to different types of LEASEQUERY. If a LEASEQUERY is sent to multiple servers, the requestor may receive from several servers lease data on the same DHCPv6 client. Additionally, if a requestor is an access concentrator, it may receive lease data from other than leasequery exchanges, e.g., [7]. This section describes how the requestor handles multiple lease data sources on the same DHCPv6 client from the same server or different servers.

The client data from the different sources may be disjoint or overlapping. The disjoint and overlapping relationship can happen between data from the same server or different servers.

If client data from two sources on the same client are of different types or values, then the data are disjoint. An example of data of different types is when a requestor receives an IPv6 address lease from one server and a prefix lease from another server, both assigned

to the same client. An example of different values (but the same type) is when a requestor receives two IPv6 address leases from two different servers, both assigned to the same client, but the leases are on two different IPv6 addresses. If the requestor receives disjoint client data from different sources, it SHOULD merge them.

If client data from two sources on the same client are of the same type and value, then the data are overlapping. An example of overlapping data is when a requestor receives a lease on the same IPv6 address from two different servers. Overlapping client data are also called conflicting data.

The requestor SHOULD use the OPTION_SERVER_RSN [5] to resolve data conflicts originated from the same server, and SHOULD accept data with the higher server-sequence-number. The requestor SHOULD use the OPTION_CLT_TIME to resolve data conflicts originated from different servers, and SHOULD accept data with most recent OPTION_CLT_TIME.

4.4. DHCPv6 Leasequery Server Behavior

A DHCPv6 server sends LEASEQUERY-REPLY messages in response to valid LEASEQUERY messages it receives to return the statefully assigned addresses, delegated prefixes, and other information about that match the query.

4.4.1. Receipt of LEASEQUERY Messages

Upon receipt of a valid LEASEQUERY message, the DHCPv6 server locates the requested client, collects data on the client, and constructs and returns a LEASEQUERY-REPLY. A LEASEQUERY message can not be used to assign, release, or otherwise modify bindings or other configuration information.

The server constructs a LEASEQUERY-REPLY message by setting the "msg-type" field to LEASEQUERY-REPLY, and copying the transaction ID from the LEASEQUERY message into the transaction-id field.

If the query-type in the OPTION_LQ_QUERY option is not a known or supported value, the server adds an OPTION_STATUS_CODE option with the UnknownQueryType status code and sends the LEASEQUERY-REPLY to the requestor. If the query-options do not contain the required options for the query-type, the server adds an OPTION_STATUS_CODE option with the MalformedQuery status code and sends the LEASEQUERY-REPLY to the client.

A server may also restrict LEASEQUERY messages, or query-types, to certain requestors. In this case, the server MAY discard the LEASEQUERY message or MAY add an OPTION_STATUS_CODE option with the

NotAllowed status code and send the LEASEQUERY-REPLY to the requestor.

If the OPTION_LQ_QUERY specified a non-zero link-address, the server MUST use the link-address to find the appropriate link for the client. Otherwise, the server uses the address from the OPTION_IAADDR option to find the appropriate link for the client.

At this point, the server uses the data in the OPTION_LQ_QUERY to initiate the query. The result of the query will be zero or one client. This will result in zero or one OPTION_CLIENT_DATA option being added to the LEASEQUERY-REPLY.

4.4.2. Constructing the Client's OPTION_CLIENT_DATA

An OPTION_CLIENT_DATA option in a LEASEQUERY-REPLY message MUST minimally contain the following data.

1. OPTION_CLIENTID
2. OPTION_IAADDR
3. OPTION_IAPREFIX
4. OPTION_CLT_TIME

Depending on the bindings the client has on a link, either OPTION_IAADDR options, OPTION_IAPREFIX options, or both may be present.

The OPTION_CLIENT_DATA SHOULD include options requested in the OPTION_ORO of the OPTION_LQ_QUERY option in the LEASEQUERY message and that are acceptable to return based on the list of "sensitive options", discussed below.

DHCPv6 servers SHOULD be configurable with a list of "sensitive options" that must not be returned to the requestor when specified in the OPTION_ORO of the OPTION_LQ_QUERY option in the LEASEQUERY message. Any option on this list MUST NOT be returned to a requestor, even if requested by that requestor.

4.4.3. Transmission of LEASEQUERY-REPLY Messages

The server sends the LEASEQUERY-REPLY message as described in the "Transmission of Reply Messages" section of [2].

5. Security Considerations

The senders of LEASEQUERY messages are expected to be within the same security domain as the DHCPv6 server. As such, the security threat to DHCPv6 leasequery is inherently an insider threat. However, this

document doesn't prohibit entities in external security domains from sending LEASEQUERY messages to DHCPv6 servers. Regardless of the network configuration, however, the potential attacks by insiders and outsiders are the same.

If the requestor is an access concentrator, DHCPv6 leasequery security SHOULD follow security between the relay agent and the DHCPv6 server as described in [2] Sections 21.1 and 22.11. Requestors are essentially a DHCPv6 client for the purposes of the LEASEQUERY message. Thus, DHCPv6 authentication [2] is also an appropriate mechanism for securing LEASEQUERY and LEASEQUERY-REPLY messages.

Access concentrators are expected to be common leasequery requestors. Access concentrators that use DHCPv6 gleaning (i.e., [7]), refreshed with LEASEQUERY messages, will maintain accurate client/binding information. This ensures that the access concentrator can forward data traffic to the intended destination in the broadband access network, can perform IPv6 source address verification of datagrams from the access network, and can encrypt traffic that can only be decrypted by the intended access modem (e.g., [BPI] and [BPI+]). Thus, the LEASEQUERY message allows an access concentrator to provide considerably enhanced security. DHCPv6 servers SHOULD prevent exposure of their information (particularly the mapping of hardware address to IPv6 address, which can be an invasion of broadband subscriber privacy) by employing the techniques detailed in [2], Section 21, "Authentication of DHCP Messages".

DHCPv6 servers SHOULD also provide for the ability to restrict the information that they make via leasequery, as described in Section 4.4.2.

DHCPv6 servers supporting LEASEQUERY SHOULD ensure that they cannot be successfully attacked by being flooded with large quantities of LEASEQUERY messages in a short time. In some environments, it may be appropriate to configure a DHCPv6 server with the IPv6 source addresses of the relay agents for which it may respond to LEASEQUERY messages, thereby allowing it to respond only to requests from only a handful of relay agents. This does not provide any true security, but may be useful to thwart unsophisticated attacks of various sorts.

Replayed messages can represent a DOS attack through exhaustion of processing resources, bogus leasequery requestors can send a lot of LEASEQUERY messages to overwhelm a DHCPv6 server, thus preventing the server from serving legitimate and regular DHCPv6 clients as well as legitimate DHCPv6 leasequery requestors, denying configurations to legitimate DHCPv6 clients as well lease information to legitimate DHCPv6 leasequery requestors.

One attack specific to an access concentrator as a requestor is the establishment of a malicious server with the intent of providing incorrect lease or route information to the access concentrator, thwarting source IPv6 address verification and preventing correct routing.

The use of the OPTION_SERVER_RSN option [5] does provide an attacker that also knows the server's DUID the ability to effectively lock out future updates from the real server by supply a large sequence number.

6. IANA Considerations

IANA is requested to assign the following new DHCPv6 Message types in the registry maintained in http://www.iana.org/assignments/dhcpv6-parameters:

```
LEASEQUERY
LEASEQUERY-REPLY
```

IANA is requested to assign the following new DHCPv6 Option Codes in the registry maintained in http://www.iana.org/assignments/dhcpv6-parameters:

```
OPTION_LQ_QUERY
OPTION_CLIENT_DATA
OPTION_CLT_TIME
```

IANA is requested to assign the following new DHCPv6 Status Codes in the registry maintained in http://www.iana.org/assignments/dhcpv6-parameters:

```
UnknownQueryType
MalformedQuery
NotConfigured
NotAllowed
```

IANA is requested to create a new registry for the OPTION_LQ_QUERY option query-type codes in the registry maintained in http://www.iana.org/assignments/dhcpv6-parameters with the following initial assignments:

```
QUERY_BY_ADDRESS       1
```

7. Acknowledgements

Thanks to Ralph Droms, Richard Johnson, Josh Littlefield, Hemant Singh, Pak Siripunkaw, Markus Stenberg, and Ole Troan for their input, ideas, and review during the production of this document.

8. References

8.1. Normative References

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

[2] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

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

[4] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[5] Volz, B. and R. Droms, "DHCPv6 Server Reply Sequence Number Option (draft-volz-dhc-dhcpv6-srsn-option-*)", August 2006.

8.2. Informative References

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

[7] Droms, R., Volz, B., and O. Troan, "DHCP Relay Agent Assignment Notification Option (draft-ietf-dhc-dhcpv6-agentopt-delegate-*)", August 2006.

Authors' Addresses

John Jason Brzozowski
Comcast Cable
1800 Bishops Gate Boulevard
Mt. Laurel, NJ 08054
USA

Phone: +1 856 324 2671
Email: john_brzozowski@cable.comcast.com

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

Phone: +1 978 936 0000
Email: kkinnear@cisco.com

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

Phone: +1 978 936 0000
Email: volz@cisco.com

Shengyou Zeng
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719
USA

Phone: +1 978 936 0000
Email: szeng@cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).