

DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 26, 2014

Q. Sun
Y. Cui
Tsinghua University
M. Siodelski
ISC
S. Krishnan
Ericsson
I. Farrer
Deutsche Telekom AG
November 22, 2013

DHCPv4 over DHCPv6 Transport
draft-ietf-dhc-dhcpv4-over-dhcpv6-03

Abstract

IPv4 connectivity is still needed as networks migrate towards IPv6. Users require IPv4 configuration even if the uplink to their service provider supports IPv6 only. This document describes a mechanism for obtaining IPv4 configuration information dynamically in IPv6 networks by carrying DHCPv4 messages over DHCPv6 transport. Two new DHCPv6 messages as well as new DHCPv6 options are defined for the purpose of conveying DHCPv4 messages through IPv6 networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on May 26, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal

Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Requirements Language	3
3.	Terminology	3
4.	Architecture Overview	3
5.	New DHCPv6 Messages	5
5.1.	Message Types	5
5.2.	Message Formats	5
5.3.	Boot-request-v6 Message Flags	6
5.4.	Boot-reply-v6 Message Flags	6
6.	New DHCPv6 Options	7
6.1.	BOOTP Message Option Format	7
6.2.	DHCPv4-over-DHCPv6 Enable Option Format	7
6.3.	4o6 Server Address Option Format	8
7.	Use of the Boot-request-v6 Unicast Flag	8
8.	4o6 DHCP Client Behavior	9
9.	Relay Agent Behavior	10
10.	4o6 DHCP Server Behavior	11
11.	Security Considerations	11
12.	IANA Considerations	12
13.	Contributors List	12
14.	References	12
14.1.	Normative References	12
14.2.	Informative References	13
	Authors' Addresses	13

[1.](#) Introduction

As the migration towards IPv6 continues, IPv6-only networks will become more prevalent. At the same time, IPv4 connectivity will continue to be provided as a service over IPv6-only networks. In addition to providing IPv4 addresses for clients of this service, other IPv4 configuration parameters may also need to be provided (e.g. addresses of IPv4-only services).

This document describes a transport mechanism to carry DHCPv4 messages using DHCPv6 protocol, for the dynamic provisioning of IPv4 addresses and other DHCPv4 specific configuration parameters across

IPv6-only networks. It leverages the existing infrastructure for DHCPv4, e.g. failover, DNS updates, leasequery, etc.

2. Requirements Language

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

3. Terminology

This document makes use of the following terms:

- 4o6 DHCP Client: A DHCP client which supports both DHCPv6 protocol [[RFC3315](#)] as well as the DHCPv4 over DHCPv6 protocol described in this document. Such a client is capable to request its IPv6 configuration using DHCPv6 and IPv4 configuration using DHCPv4 over DHCPv6.
- 4o6 DHCP Server: A DHCP server that is capable of processing DHCPv4 packets encapsulated in the BOOTP Message option (defined below).
- CPE: Customer Premises Equipment (also known as Customer Provided Equipment), which provides the access of devices connected to Local Area Network (typically at customer's site/home) to Internet Service Provider's network.
- DHCPv4 over DHCPv6: A protocol described in this document, which is used to carry DHCPv4 messages in the payload of DHCPv6 messages.

4. Architecture Overview

The architecture described in this document addresses a typical use case, where a DHCP client's uplink supports IPv6 only and the Service Provider's network supports IPv6 and limited IPv4 services. In this scenario, the client can only use the IPv6 network to access IPv4 services. So it must configure IPv4 services using IPv6 as the underlying network protocol.

Although the purpose of this document is to address the problem of communication between the DHCPv4 client and the DHCPv4 server, the mechanism that it describes does not restrict the transported messages types only to DHCPv4. As the DHCPv4 message is a special type of the BOOTP message, BOOTP messages can also be transported

using the same mechanism.

DHCP clients can be running on CPE devices, end hosts or any other device which supports the DHCP client function. At the time of writing, DHCP clients on CPE devices are easier to modify compared to those implemented on end hosts. As a result, this document uses the CPE as an example for describing the mechanism. This does not preclude any end-host, or other device requiring IPv4 configuration, from implementing the mechanism in the future.

This mechanism works by carrying DHCPv4 messages encapsulated within DHCPv6 messages. Figure 1, below, illustrates one possible deployment architecture.

The 4o6 DHCP client implements a new DHCPv6 message called Boot-request-v6, which contains a new option called BOOTP Message option. The format of this option is described in [Section 6.1](#).

The DHCPv6 message can be transmitted either via DHCPv6 Relay Agents or directly to the 4o6 DHCP Server. The server replies with a Boot-reply-v6 message, which is a new DHCPv6 message type. This message carries the DHCPv4 response encapsulated in the BOOTP Message option.

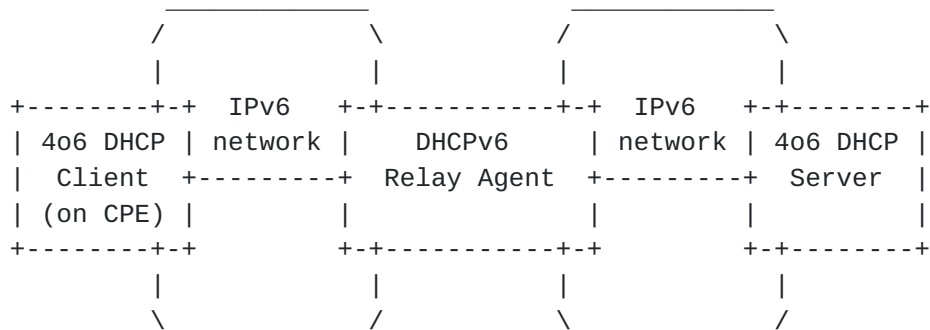


Figure 1: Architecture Overview

By default, the DHCPv4-over-DHCPv6 function MUST be disabled on the client. Before the client can use DHCPv4 over DHCPv6, it MUST obtain the IPv6 configuration. It requests the DHCPv4-over-DHCPv6 Enable option by sending its code in Option Request Option (ORO) described in [\[RFC3315\]](#). The DHCPv6 server includes the DHCPv4-over-DHCPv6 Enable option in response to a client's request to instruct the client to use DHCPv4 over DHCPv6 for IPv4 configuration. The format of the DHCPv4-over-DHCPv6 Enable option is described in [Section 6.2](#).

Typically, a 4o6 DHCP client communicates with the 4o6 DHCP servers using well-known All_DHCP_Relay_Agents_and_Servers multicast address.

Client SHOULD request the 4o6 Server Address Option from a DHCPv6 server and the server may be configured to respond to the client with one such option that contains one or more unicast addresses of the 4o6 DHCP Servers. The server includes 4o6 Server Address Option in Advertise and Reply messages. The format of the option is defined in [Section 6.3](#).

5. New DHCPv6 Messages

There are two new DHCPv6 messages defined in this document which carry DHCPv4 messages between a client and a server using DHCPv6 protocol: Boot-request-v6 and Boot-reply-v6. This section describes the structures of these messages.

5.1. Message Types

BOOTREQUESTV6 (TBD): Identifies a Boot-request-v6 message. A 4o6 DHCP client sends this message to a 4o6 DHCP server. The BOOTP Message Option carried by this message contains a BOOTREQUEST message that the 4o6 DHCP client uses to request IPv4 configuration parameters from the server.

BOOTREPLYV6 (TBD): Identifies a Boot-reply-v6 message. A 4o6 DHCP server sends this message to a 4o6 DHCP client. It contains a BOOTP Message Option carrying a BOOTREPLY message in response to a BOOTREQUEST received by the server in the BOOTP Message Option of the Boot-request-v6 message.

5.2. Message Formats

Both DHCPv6 messages defined in this document share the following format:

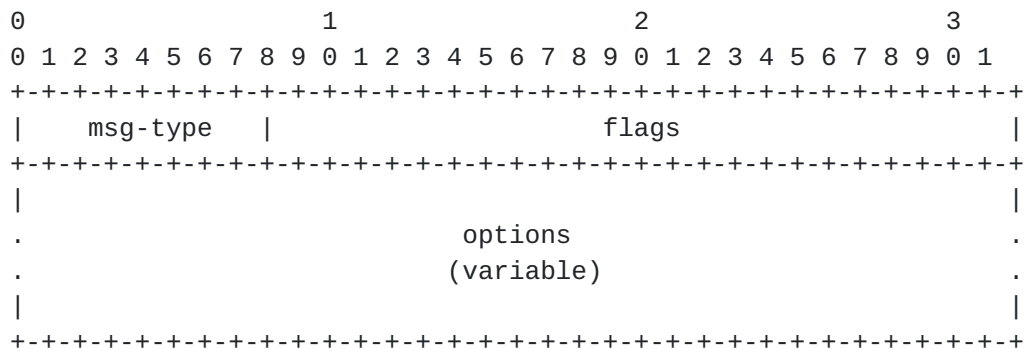


Figure 2: Architecture Overview

msg-type	Identifies message type. It can be either BOOTREQUESTV6 (TBD) or BOOTREPLYV6 (TBD) which corresponds to the Boot-request-v6 or Boot-reply-v6, respectively.
flags	Specifies flags which provide additional information required by the server to process a DHCPv4 message encapsulated in Boot-request-v6 message, or required by the client to process DHCPv4 message encapsulated in Boot-reply-v6 message.
options	The options carried by the message. The BOOTP Message Option described in Section 6.1 MUST be carried by the message.

5.3. Boot-request-v6 Message Flags

The "flags" field of the Boot-request-v6 is used to carry additional information which may be used by the server to process the encapsulated DHCPv4 message. Currently only one bit of this field is used. Remaining bits are reserved for the future use. The "flags" field has the following format:

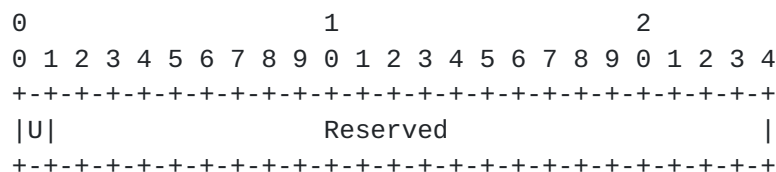


Figure 3: Boot-request-v6 flags format

U	Unicast Flag. If set to 1, it indicates that the DHCPv4 message encapsulated with the Boot-request-v6 message would be sent to a unicast address if it was sent using IPv4. If this flag is set to 0, it indicates that the DHCPv4 message would be sent to broadcast address if it was sent using IPv4.
Reserved	Bits reserved for future use. A client that doesn't implement future extensions using these bits MUST set them to 0.

5.4. Boot-reply-v6 Message Flags

This document introduces no flags to be carried in the "flags" field of the Boot-reply-v6 message. They are all reserved for the future use. The 406 Server MUST set all bits of this field to 0 and the 406 client MUST ignore the content in this field.

6. New DHCPv6 Options

6.1. BOOTP Message Option Format

The BOOTP Message option carries a BOOTP message that is sent by the client or the server. Such BOOTP messages exclude any IP or UDP headers.

The format of the BOOTP Message Option is:

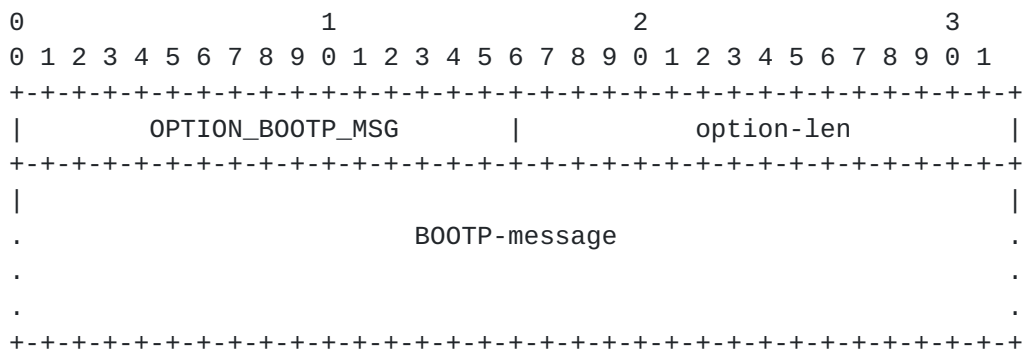


Figure 4: BOOTP Message Option Format

option-code OPTION_BOOTP_MSG (TBD).

option-len Length of BOOTP message.

BOOTP-message The BOOTP message sent by the client or the server. In a Boot-request-v6 message it contains a BOOTREQUEST message sent by a client. In a Boot-reply-v6 message it contains a BOOTREPLY message sent by a server in response to a client.

6.2. DHCPv4-over-DHCPv6 Enable Option Format

The DHCPv4-over-DHCPv6 Enable option is sent by the DHCPv6-only server to signal that the client SHOULD use DHCPv4 over DHCPv6 to obtain IPv4 configuration. The server includes this option if it is requested by the client.

The format of the DHCPv4-over-DHCPv6 Enable option is:

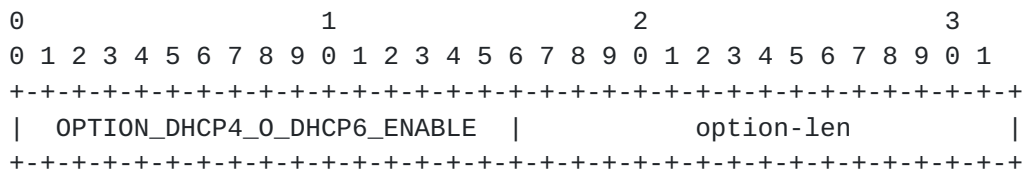


Figure 5: DHCPv4-over-DHCPv6 Enable Option Format

```

option-code    OPTION_DHCP4_O_DHCP6_ENABLE (TBD).

option-len     0

```

6.3. 4o6 Server Address Option Format

The 4o6 Server Address option carries one or more unicast IPv6 addresses of the 4o6 DHCP Server(s). The DHCPv6-only server includes this option if it is requested by the client.

The format of the 4o6 Server Address option is:

```

      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_DHCP4_O_DHCP6_SERVER |          option-len          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|
|                               IPv6 Address(es)
|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Figure 6: 4o6 Servers Address Option Format

```

option-code    OPTION_DHCP4_O_DHCP6_SERVER (TBD).

option-len     Length of the IPv6 address(es) carried by the option,
                  i.e. multiple of 16 octets.

IPv6 Address   One or more IPv6 addresses of the 4o6 DHCP Server(s).

```

7. Use of the Boot-request-v6 Unicast Flag

A DHCPv4 client conforming to the [RFC2131] may send its DHCPREQUEST message to either broadcast or unicast address depending on its state. For example, the client in the RENEWING state uses a unicast address to contact a DHCPv4 server to renew its lease. The client in the REBINDING state uses a broadcast address. If there is a DHCPv4 relay agent in the middle, a client in the RENEWING state may send a DHCPREQUEST message to the unicast address of the relay agent. In such case the server can't find out whether the client sent a message to a unicast or broadcast address and thus it can't determine the client's state. [RFC5010] introduced the "Flags Suboption" which relay agents add to relayed messages to indicate whether broadcast or

unicast was used by the client.

In DHCPv4 over DHCPv6, IPv6 is used to deliver DHCPv4 messages to the 4o6 DHCP Server. There is no relation between the outer IPv6 address and the inner DHCPv4 message. So the server is not able to know whether the DHCPv4 messages should have been sent using broadcast or unicast in IPv4 by checking the IPv6 address. This is similar to the case addressed by the [[RFC5010](#)].

In order to allow the server to determine the client's state, the "Unicast" flag is carried in the Boot-request-v6 message. Client MUST set this flag to 1 when the DHCPv4 message would have been sent to the unicast address if using DHCPv4 over IPv4. This flag MUST be set to 0 if the DHCPv4 client would have sent the message to the broadcast address in IPv4. The choice whether a given message should be sent to a broadcast or unicast address MUST be made based on the [[RFC2131](#)] and its extensions.

8. 4o6 DHCP Client Behavior

The DHCPv4-over-DHCPv6 function MUST be disabled by default. The client MUST obtain its IPv6 configuration before using DHCPv4 over DHCPv6. The client that intends to use DHCPv4 over DHCPv6 MUST request the DHCPv4-over-DHCPv6 Enable Option and SHOULD request the 4o6 Server Address Option in the Option Request Option (ORO) in every Solicit, Request, Renew and Information-request messages. The 4o6 DHCP client MUST NOT request the DHCPv4-over-DHCPv6 Enable Option nor the 4o6 Server Address Option in the Boot-request-v6 message.

The DHCPv6 server MAY include these options in the responses to the client. The client determines how to enable the DHCPv4-over-DHCPv6 function based on the presence / absence of the two options:

- o If the client doesn't receive the DHCPv4-over-DHCPv6 Enable option, it MUST NOT enable the DHCPv4-over-DHCPv6 function. In the case where the DHCPv4 over DHCPv6 service is running, the client MUST disable the function.
- o If the client receives the DHCPv4-over-DHCPv6 Enable Option but no 4o6 Servers Address Option, it SHOULD enable the DHCPv4-over-DHCPv6 function and use IPv6 All_DHCP_Relay_Agents_and_Servers multicast address to communicate with servers and relays.
- o If the client receives both options, it SHOULD enable the DHCPv4-over-DHCPv6 function and send requests to the unicast address(es) in the 4o6 Server Address Option.
- o If the client only receives 4o6 Server Address Option, the client

MUST ignore the 4o6 Server Address Option and MUST NOT enable the DHCPv4-over-DHCPv6 function.

The client supporting DHCPv4 over DHCPv6 SHOULD use Information Refresh Time Option [[RFC4242](#)] to refresh the status of DHCPv4-over-DHCPv6 service as well as other DHCPv6 configuration data.

The client signaled by the server to use DHCPv4 over DHCPv6 SHOULD cease to send DHCPv4 messages using DHCP protocol described in [[RFC2131](#)] and use the DHCPv4 over DHCPv6 to request IPv4 configuration from the 4o6 DHCP Server. The DHCPv4 message is stored verbatim in the BOOTP Message option carried by the Boot-request-v6 message. The client MUST put exactly one BOOTP Message option into a single Boot-request-v6 message.

Client MUST follow rules defined in [Section 7](#) when setting Unicast flag.

If the client has not received the 4o6 Server Addresses option from the DHCPv6 server, it transmits the Boot-request-v6 message as specified in [Section 13 of \[RFC3315\]](#). If the client received this option, it SHOULD send Boot-request-v6 message to all unicast addresses listed in the option.

On receiving a Boot-reply-v6 message, the client MUST look for the BOOTP Message option within this message. If this option is not found, the Boot-reply-v6 message is discarded. If the BOOTP Message Option presents, the client extracts the DHCPv4 message it contains and processes it as described in [section 4.4 of \[RFC2131\]](#).

When dealing with IPv4 configuration, the 4o6 DHCP client SHOULD follow the normal DHCPv4 retransmission requirements and strategy as specified in [section 4.1 of \[RFC2131\]](#). There are no explicit transmission parameters associated with a Boot-request-v6 message.

The 4o6 DHCP client MUST implement [[RFC4361](#)] to ensure that the device correctly identifies itself.

9. Relay Agent Behavior

When a DHCPv6 relay agent receives a Boot-request-v6 message, it may not recognize this message. It can just forward this message as in [[I-D.ietf-dhc-dhcpv6-unknown-msg](#)].

Additionally, the DHCPv6 relay agent MAY allow the configuration of a dedicated DHCPv4 over DHCPv6 specific destination address(es), differing from the address(es) of the DHCPv6-only server(s). To implement this function, the relay checks the received DHCPv6 message

type and forwards according to the following logic:

1. If the message type is BOOTREQUESTV6, the packet is relayed to the configured 4o6 DHCP Server's address(es) in the form of normal DHCPv6 packet (i.e. DHCPv6/UDP/IPv6).
2. For any other DHCPv6 message type, forward according to [section 20 of \[RFC3315\]](#).

The above logic only allows for separate relay destinations configured on the relay agent closest to the client (single relay hop). Multiple relaying hops are not considered in the case of separate relay destinations.

[10.](#) 4o6 DHCP Server Behavior

When the server receives a Boot-request-v6 message from a client, it searches for the BOOTP Message Option. The server discards the packet without this option. The server MAY notify an administrator about the receipt of a malformed packet. The mechanism for this notification is out of scope for this document

If the server finds a valid BOOTP Message option, it extracts the original DHCPv4 message and the contents of the "flags" field carried in the Boot-request-v6 message and uses them to generate the appropriate DHCPv4 response (server to client message). The response is generated as described in [\[RFC2131\]](#) with the exception that the server SHOULD use the information carried in the "flags" field of the Boot-request-v6 message to find out whether the client's message would have been sent to the broadcast or unicast address if DHCPv4 protocol was used. This is useful for the server to determine the state of the client. The use of the "flags" field is described in detail in [Section 7](#).

When appropriate DHCPv4 response is generated, the 4o6 Server places it in the payload of a BOOTP Message Option, which it puts into the Boot-reply-v6 message.

If the Boot-request-v6 message was received directly by the server, the Boot-reply-v6 message MUST be unicast from the interface on which the original message was received.

If the Boot-request-v6 message was received in a Relay-forward message, the server creates a Relay-reply message with the Boot-reply-v6 message in the payload of a Relay Message option, and responds as described in [section 20.3 of \[RFC3315\]](#).

[11.](#) Security Considerations

In this specification, DHCPv4 messages are encapsulated in the newly defined option and messages. This is similar to the handling of the current relay agent messages. In order to bypass firewalls or network authentication gateways, a malicious attacker may leverage this feature to convey other messages using DHCPv6, i.e. use DHCPv6 as a form of encapsulation. However, the potential risk from this is no more severe than that with the current DHCPv4 and DHCPv6 practice.

There are chances that a rogue DHCPv6 server may reply with a 4o6 Server Address Option containing duplicated unicast IPv6 addresses, which can cause an amplification attack. To avoid this, the client MUST check if there are repeated IPv6 addresses in a 4o6 Server Address Option when receiving one. The client MUST ignore those duplicated unicast IPv6 addresses.

12. IANA Considerations

IANA is requested to allocate three DHCPv6 option codes for use by OPTION_BOOTP_MSG, OPTION_DHCP4_O_DHCP6_ENABLE and OPTION_DHCP4_O_DHCP6_SERVERS, and two DHCPv6 message type codes for the BOOTREQUESTV6 and BOOTREPLYV6.

13. Contributors List

Many thanks to Ted Lemon, Bernie Volz, Tomek Mrugalski, Yuchi Chen and Cong Liu, for their great contributions to the draft.

14. References

14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3315](#), July 2003.
- [RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 4242](#), November 2005.
- [RFC4361] Lemon, T. and B. Sommerfeld, "Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)", [RFC 4361](#), February 2006.

14.2. Informative References

- [I-D.ietf-dhc-dhcpv6-unknown-msg]
Cui, Y., Sun, Q., and T. Lemon, "Handling Unknown DHCPv6 Messages", [draft-ietf-dhc-dhcpv6-unknown-msg-03](#) (work in progress), November 2013.
- [RFC5010] Kinnear, K., Normoyle, M., and M. Stapp, "The Dynamic Host Configuration Protocol Version 4 (DHCPv4) Relay Agent Flags Suboption", [RFC 5010](#), September 2007.

Authors' Addresses

Qi Sun
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: sunqi@csnet1.cs.tsinghua.edu.cn

Yong Cui
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6260-3059
Email: yong@csnet1.cs.tsinghua.edu.cn

Marcin Siodelski
950 Charter Street
Redwood City, CA 94063
USA

Phone: +1 650 423 1431
Email: msiodelski@gmail.com

Suresh Krishnan
Ericsson

Email: suresh.krishnan@ericsson.com

Ian Farrer
Deutsche Telekom AG
GTN-FM4, Landgrabenweg 151
Bonn, NRW 53227
Germany

Email: ian.farrer@telekom.de