DHC Working Group   Q. Sun
Internet-Draft   Y. Cui
Intended status: Standards Track   Tsinghua University
Expires: January 16, 2014   M. Siodelski
                                                                 ISC
                                                         S. Krishnan
                                                            Ericsson
                                                           I. Farrer
                                               Deutsche Telekom AG
                                                      July 15, 2013

DHCPv4 over DHCPv6 Transport
draft-ietf-dhc-dhcpv4-over-dhcpv6-01

Abstract

   This document describes a mechanism for obtaining IPv4 address and
   other parameters in IPv6 networks by carrying DHCPv4 messages over
   DHCPv6 transport.

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 January 16, 2014.

Table of Contents

1. Introduction

As the migration towards IPv6 continues, IPv6 only networks will
become more prevalent. However, IPv4 connectivity will continue to
be provided as a service over these 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).

By conveying DHCPv4 messages over DHCPv6 transport, this mechanism
can achieve dynamic provisioning of IPv4 address and other
configuration parameters. In addition, it is able to leverage
existing infrastructure for DHCPv4, e.g. failover, DNS updates,
leasequery, etc. This mechanism is suitable for stateful allocation
and management of IPv4 addresses and configuration parameters across
IPv6 networks.

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:

DHCP client: The 'DHCP client' in this document consists of
                        both DHCPv4 and DHCPv6 client engines. The
                        client is able to request IPv6 information
                        through DHCPv6, as well as to request IPv4
                        information using DHCPv4-over-DHCPv6 transport.

4o6 Server: A DHCP server capable of processing DHCPv4
                        packets wrapped in the BOOTP Message Option
                        (defined below).

DHCPv4-over-DHCPv6:   A protocol described in this document, which is
                        used to carry DHCPv4 messages encapsulated in
                        DHCPv6 messages.

4. New DHCPv6 Messages

The following new DHCPv6 Client/Server messages are defined by this
document. These are formatted as specified in Section 6 of
 [RFC3315].

BOOTREQUESTV6 (TBD): A client sends a BOOTREQUESTV6 message to a
                      server, which contains a BOOTP Message Option.
                      The BOOTP Message Option contains a BOOTREQUEST
                      message that the client uses to request IPv4
                      configuration parameters from the server.

BOOTREPLYV6 (TBD):   A server sends a BOOTREPLYV6 message containing
                      a BOOTP Message Option in response to a
                      client's BOOTREQUESTV6 message. The BOOTP
                      Message Option contains a BOOTREPLY message in
                      response to a BOOTREQUEST received by the
                      server in the BOOTP Message Option of a
                      BOOTREQUESTV6 message.

5. Architecture Overview

The architecture described in this document addresses a typical use
case, whereby 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 and so it must configure IPv4 services using IPv6 as
the underlying transport protocol.

Although the purpose of this document is to address the problem of
communication between DHCPv4 client and DHCPv4 server, the mechanism
it describes does not restrict the transported types of messages to
DHCPv4. BOOTP messages can 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 relatively 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 doesn't preclude end hosts from implementing the mechanism in
the future.

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

The DHCP client implements a new DHCPv6 message called BOOTREQUESTV6,
which contains a new option called BOOTP Message Option. The format
of the option is described in Section 6.1.

The DHCPv6 packet can be transmitted either via Relay Agents or
directly to the 4o6 Server. The server replies with a relevant
DHCPv6 packet carrying DHCPv4 response wrapped with the BOOTP Message
Option.

```
               _____            _____
              /              \          /              \
              |              |          |              |
         +--------+-+  IPv6   +-+------------+-+  IPv6   +-+--------+
         |  DHCP   | network |     DHCP       | network |   4o6    |
         | Client  +---------+  Relay Agent   +---------+  Server  |
         | (on CPE) |         |               |         |          |
         +--------+-+         +-+------------+-+         +-+--------+
              |              |          |              |
              _____/          _____/
```

                   Figure 1: Architecture Overview

The DHCPv4-over-DHCPv6 is by default disabled on the client. Before
client can use this protocol it MUST obtain configuration using
DHCPv6 as described in [RFC3315]. During this configuration, server
MAY include DHCPv4-over-DHCPv6 Enable Option in its Reply message to
indicate that client SHOULD use DHCPv4-over-DHCPv6 protocol to obtain
additional configuration. The format of the DHCPv4-over-DHCPv6
Enable Option is described in Section 6.2

Typically, client communicates with the 4o6 Servers using well known
All_DHCP_Relay_Agents_and_Servers multicast address. If DHCPv6
server is configured to do so, it MAY send unicast addresses of the
4o6 Servers to the client during client's configuration using DHCPv6.
The unicast addresses are carried in the 4o6 Server Addresses Option
encapsulated in the Reply message. The 4o6 Server Addresses Option's
format is defined in Section 6.3.

6. 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:

```
  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_BOOTP_MSG        |           option-len          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 .                          BOOTP-message                        .
 .                                                               .
 .                                                               .
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

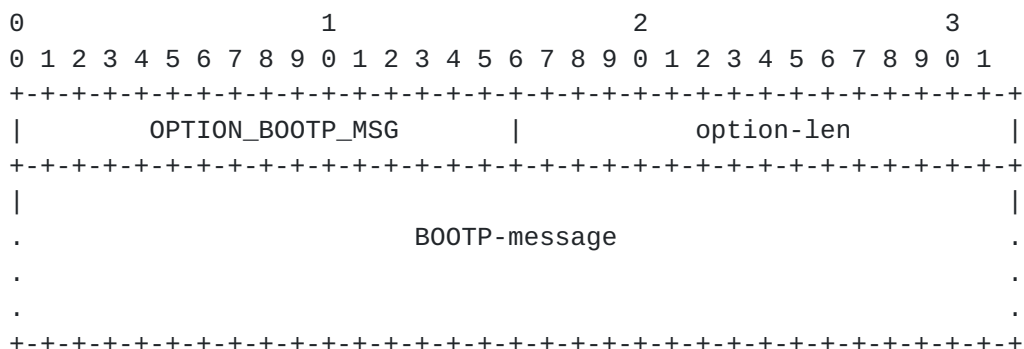   Figure 2: BOOTP Message Option Format

   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 BOOTREQUESTV6 message it contains a BOOTREQUEST
                   message sent by client. In a BOOTREPLYV6 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 indicates that the client SHOULD
enable the DHCPv4-over-DHCPv6 function.
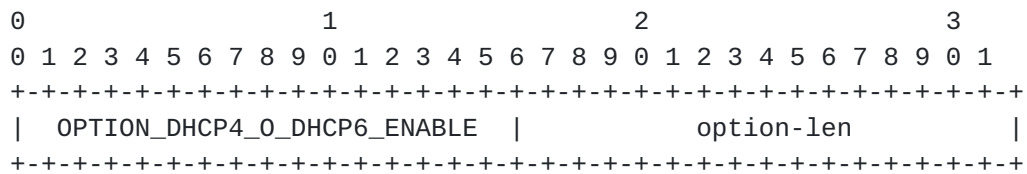
The format of the DHCPv4-over-DHCPv6 Enable Option is:

```
 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_ENABLE  |             option-len        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

             Figure 3: DHCPv4-over-DHCPv6 Enable Option Format

   option-code   OPTION_DHCP4_O_DHCP6_ENABLE (TBD)

   option-len   0

6.3. 4o6 Servers Address Option Format

   The 4o6 Servers Address Option carries unicast IPv6 addresses of the
   4o6 Servers.

   The format of the 4o6 Servers 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_SERVERS  |             option-len        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                         IPv6 Address(es)                      .
.                                                               .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
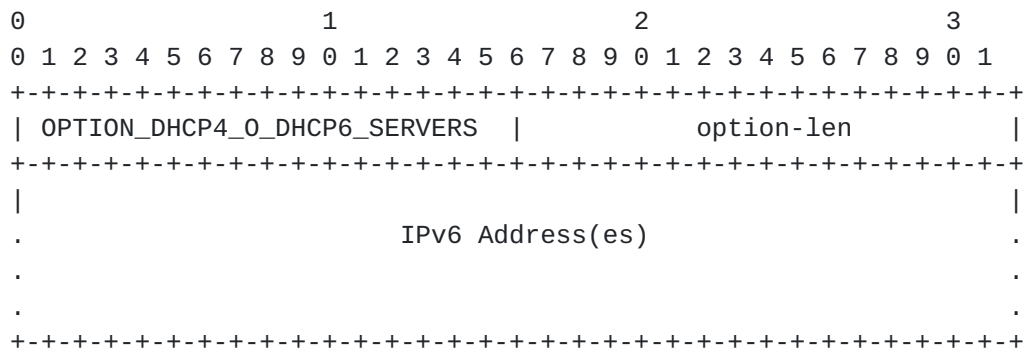
               Figure 4: 4o6 Servers Address Option Format

   option-code   OPTION_DHCP4_O_DHCP6_SERVERS (TBD)

   option-len   Length of the IPv6 address(es), i.e. integer times
                   of 16.

   IPv6 Address   The IPv6 address(es) of the 4o6 Server(s).

7. Client Behavior

   The DHCP client SHOULD request the DHCPv4-over-DHCPv6 Enable Option
   and the 4o6 Server Addresses Option in the Option Request Option
   (ORO) to launch the DHCPv4-over-DHCPv6 function.

   Client MUST NOT initiate communication with 4o6 Servers before it

obtains configuration using DHCPv6 as described in [RFC3315]. If
client supports DHCPv4-over-DHCPv6 function it SHOULD request the
DHCPv4-over-DHCPv6 Enable Option and 4o6 Server Addresses Option in
the Option Request Option (ORO). DHCPv6 server MAY include these
options in Reply message sent to the client. The client determines
how to launch the DHCPv4-over-DHCPv6 function using the presence /
absence of these two options:

o If the client doesn't receive the DHCPv4-over-DHCPv6 Enable
   Option, it SHOULD NOT enable the DHCPv4 over DHCPv6 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, but use IPv6 multicast to communicate with the
   servers or relays as described above.

o If the client receives both options, it SHOULD enable the DHCPv4
   -over-DHCPv6 function, and send requests to all unicast addresses
   conveyed by the 4o6 Server Addresses Option.

If client is instructed by the DHCPv6 server to use DHCPv4-over-
DHCPv6 function it MUST generate a DHCPv4 message to obtain
configuration from the 4o6 Server. This message is stored verbatim
in the BOOTP Message Option carried by the BOOTREQUESTV6 message.
Client MUST put exactly one BOOTP Message Option into a single
BOOTREQUESTV6 message.

If client did not receive a 4o6 Server Addresses Option from the
DHCPv6 server, it transmits the BOOTREQUESTV6 message as specified in
Section 13 of [RFC3315]. If client received this option it MUST send
BOOTREQUESTV6 message to all unicast addresses listed in the received
option.

When a client receives a BOOTREPLYV6 message, it MUST look for the
BOOTP Message Option within this message. If this option is not
found, the BOOTREPLYV6 message is discarded. If the BOOTP Message
Option is found, the client extracts the DHCPv4 message it contains
and processes it as described in section 4.4 of [RFC2131].

DHCP clients are responsible for the retransmission of messages.
When requesting IPv4 information, the client SHOULD follow the normal
DHCPv4 retransmission requirements and strategy as specified in
section 4.1 of [RFC2131]. As a result there are no explicit
transmission parameters associated with a BOOTPREQUESTV6 message.

As the DHCPv4 and DHCPv6 clients are running on the same host, the
client MUST implement [RFC4361] to ensure that the device correctly
identifies itself.

The IPv4 address allocated from the server MAY be assigned to a
different interface from the IPv6 interface requesting the
information. Future documents depending on this memo MUST specify
which IPv6 interface is to be used by the client for that purpose.

8. **Relay Agent Behavior**

When a DHCPv6 relay agent receives a BOOTREQUESTV6 message, it MUST
handle the message as described in section 4 of
[I-D.ietf-dhc-dhcpv6-unknown-msg].

A DHCPv6 relay agent MUST implement the Relay behaviour described in
section 20.1.1 of [RFC3315].

Additionally, the DHCPv6 relay agent MAY allow the configuration of
dedicated DHCPv4 over DHCPv6 specific destination addresses,
differing from the addresses 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, then the DHCPv6 request is
    relayed to the configured DHCPv4 aware 4o6 Server's address(es).

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.

9. **Server Behavior**

When server receives a BOOTREQUESTV6 message from a client, it
searches for a BOOTP Message Option. If this option is missing, the
server discards the packet. 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 sent by the client. This message is passed
to the DHCPv4 server engine, which generates a response to the client
as specified in [RFC2131]. This engine can be implemented as a
built-in DHCPv4 server function of the 4o6 Server, or it can be a
separate DHCPv4 server instance. Discussion regarding communication
between the 4o6 Server and a DHCPv4 server engine is out of scope for
this document.

When appropriate DHCPv4 response is generated, 4o6 Server places it
in the payload of a BOOTP Message Option, which it puts into the
BOOTREPLYV6 message.

If the BOOTREQUESTV6 message was received directly by the server, the
BOOTREPLYV6 message MUST be unicast from the interface on which the
original message was received.

If the BOOTREQUESTV6 message was received in a Relay-forward message,
the server creates a Relay-reply message with the BOOTREPLYV6 message
in the payload of a Relay Message Option, and responds as described
in section 20.3 of [RFC3315].

10. Security Considerations

In this specification, DHCPv4 messages are encapsulated in the newly
defined option and messages. This is similar to handling 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
greater than that with current DHCPv4 and DHCPv6 practice.

11. IANA Considerations

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

12. Contributors List

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

13. References

13.1. Normative 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-01 (work in
          progress), June 2013.

[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.

   [RFC4361] Lemon, T. and B. Sommerfeld, "Node-specific Client
              Identifiers for Dynamic Host Configuration Protocol
              Version Four (DHCPv4)", RFC 4361, February 2006.

13.2. Informative References

   [I-D.ietf-dhc-dhcpv4-over-ipv6]
              Cui, Y., Wu, P., Wu, J., and T. Lemon, "DHCPv4 over IPv6
              Transport", draft-ietf-dhc-dhcpv4-over-ipv6-06 (work in
              progress), March 2013.

Authors' Addresses

   Qi Sun
   Tsinghua University
   Department of Computer Science, Tsinghua University
   Beijing 100084
   P.R.China

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

   Yong Cui
   Tsinghua University
   Department of Computer Science, 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