DHC Working Group Q. Sun
Internet-Draft Y. Cui
Intended status: Standards Track Tsinghua University
Expires: October 28, 2013 M. Siodelski
ISC
S. Krishnan
Ericsson
I. Farrer
Deutsche Telekom AG
April 26, 2013

DHCPv4 over DHCPv6 Transport
draft-ietf-dhc-dhcpv4-over-dhcpv6-00

Abstract

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

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.

Several approaches for provisioning such information have already been proposed. This document describes alternative approach.

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:

BOOTREQUESTV6 (TBD): A new type of DHCPv6 Client/Server message defined in this document. A client sends a BOOTREQUESTV6 message to a server, which contains a BOOTP Message Option. Each BOOTP Message Option contains a BOOTREQUEST message that the client uses to request IPv4 configuration parameters from the server.

BOOTREPLYV6 (TBD): A new type of DHCPv6 Client/Server message defined in this document. A server sends a BOOTREPLYV6 message containing a BOOTP Message Option in response to a client's BOOTREQUESTV6 message. Each BOOTP Message Option, wrapped in a BOOTREPLYV6 message, contains a BOOTREPLY message. This contains the BOOTREQUEST response corresponding to a client's BOOTREQUESTV6 message.

4. 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 5. The client sends all DHCPv6 packets, including DHCPv4 over DHCPv6 packets, to the well-known All_DHCP_Relay_Agents_and_Servers multicast address on the DHCPv6 server port (UDP port 547).

The DHCPv6 packet can be transmitted either via Relay Agents or directly to the server. The server is referred in this document as a "Unified Server" for its capability of processing regular DHCPv6 traffic as well as DHCPv4 packets wrapped in the BOOTP Message Option. Server replies with a relevant DHCPv6 packet carrying DHCPv4 response wrapped with the BOOTP Message Option. Clients receive a response on UDP port 546.

```
                  _____________             _____________
                 /             \           /             \
                 |             |           |             |
        +--------+-+  IPv6   +-+-----------+-+  IPv6   +-+--------+
        |   DHCP   | network |     DHCP      | network | Unified  |
        |  Client  +---------+  Relay Agent  +---------+  Server  |
        | (on CPE) |         |               |         |          |
        +--------+-+         +-+-----------+-+         +-+--------+
                 |             |           |             |
                 \_____________/           \_____________/
```

Figure 1: Architecture Overview

5. 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. Client Behavior

When a client requires an IPv4 address and/or other IPv4 configuration parameters, it MUST generate a DHCPv4 message to obtain them from a DHCP server. This message is stored verbatim in the BOOTP Message Option carried by the BOOTREQUESTV6 message. A Client MUST put exactly one BOOTP Message Option into a single BOOTREQUESTV6 message. The Client sends out the BOOTREQUESTV6 message to the Well-Known multicast address, i.e. All_DHCP_Relay_Agents_and_Servers on multicast address defined in [RFC3315].

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].

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.

7. Relay Agent Behavior

When a DHCPv6 relay agent receives a BOOTREQUESTV6 message, it MUST handle the message as described in section 20.1.1 of [RFC3315].

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 a 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 unified 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 this document.

8. 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 Unified Server, or it can be a separate DHCPv4 server instance. Discussion regarding communication between the Unified Server and a DHCPv4 server engine is out of scope for this document.

When appropriate DHCPv4 respose is generated, Unified 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. This is analogous to other types of DHCPv6 messages as described in [RFC3315]. The server unicasts the Relay-reply message directly to the IP address of the relay agent from which the Relay-forward message was received.

9. 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.

10. IANA Considerations

IANA is kindly requested to allocate one DHCPv6 option code to the OPTION_BOOTP_MSG and two DHCPv6 message type codes to the BOOTREQUESTV6 and BOOTREPLYV6.

11. Contributors List

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

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

[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.

12.2. 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