DHCPv4 Behavior over IP-IP tunnel
draft-cui-softwire-dhcp-over-tunnel-00

Abstract

This document analyzes the situation where DHCPv4 is performed over IP-IP tunnel, and proposes methods to keep DHCP working under such situation. The main issue is encapsulation of DHCP packets on server side, and there’re both in-protocol solution and out-of-protocol solutions for this issue. The in-protocol solution is to have DHCP carrying the encapsulation address information, and the out-of-protocol solution is to have the DHCP server keeping track of the address mapping by inspecting DHCP packet.

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 December 4, 2011.

Copyright Notice

Copyright (c) 2011 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
Table of Contents

1. Introduction .............................................. 3
2. Problem analysis .......................................... 4
3. In-protocol and out-of-protocol solution ................... 6
   3.1. Address mapping with session id .................... 6
   3.2. Leveraging relay agent option ...................... 7
4. Acknowledgement ........................................... 8
5. References ............................................... 9
   5.1. Normative References ................................ 9
   5.2. Informative References .............................. 9
Authors’ Addresses ........................................ 10
1. Introduction

The DHCP protocol wasn’t designed with tunnel environment consideration. However, due to the development of tunnel-related mechanisms, the demand to apply DHCP in tunnel environment arises, especially in IPv6 transition scope. A typical application scenario is IP-IP Hub and spoke tunnel [RFC4925]. In this type of scenario, IP-IP tunnel is used to provide non-native IP connectivity to clients, across the heterogenous transit. If the non-native IP addresses of the clients are provided by the concentrator side, this address provisioning needs to cross the heterogeneous transit, too.

One transition mechanism that requires DHCP over tunnel is Public 4over6 [I-D.cui-softwire-host-4over6]. In this mechanism, users in IPv6 network get IPv4 access by IPv4-in-IPv6 tunnel with 4over6 concentrator. Every user employs a public IPv4 address to get full bidirectional IPv4 communication. This IPv4 address is allocated by the ISP over the IPv6 network. The draft suggests to achieve this by tunneling DHCPv4 between the 4over6 initiator (DHCPv4 client) and 4over6 concentrator (DHCPv4 server).
2. Problem analysis

The scenario of DHCPv4 over IP-IP tunnel is shown in Figure 1. DHCPv4 client and DHCPv4 server (could be a relay) are separated by an IPv6 or IPv4 network, with no DHCP relay in the middle. The DHCP DISCOVER and DHCP OFFER packets cannot reach the other end since they are broadcast packets, so a tunnel between the client and server is required to build a virtual link. Besides, when the middle network is IPv6, all DHCPv4 packets can not go through the network since they are IPv4 packets.

For the above reasons, we need to build the whole DHCP procedure on an IP-IP tunnel. The client (tunnel initiator) and server (tunnel concentrator) will encapsulate the E-IP (External-IP, IPv4) DHCP packets into I-IP (Internal-IP, could be IPv4 or IPv6) before sending them to remote end; the remote end (server or client) will decapsulate the packets to get the original E-IP DHCP packet before handing them to the DHCP process. The encapsulation on the client is natural: the client will use the server’s I-IP address as encapsulation destination address, which is usually known beforehand. The problem is the encapsulation on the server. The server serves more than one clients, and it must send every DHCP packet to the right client, each with different I-IP address.

We can see that regular data packet encapsulation on the concentrator faces the similar problem. The solution is to have the concentrator maintaining the mapping between each initiator’s E-IP address and I-IP address. When the concentrator performs encapsulation, it’ll use the packet’s E-IP destination address to look up the I-IP encapsulation destination address. However, this solution doesn’t
apply to DHCP packets, because the address mapping can only be established after the DHCP address allocation, and also because the destination address of DHCPOFFER packet are broadcast address. So we need some extra effort to make the encapsulation of DHCP packet work, i.e., make the concentrator encapsulate each DHCP packet with the I-IP address of the right initiator and hence send them to the right initiator.
3. In-protocol and out-of-protocol solution

So far we’ve come to two solutions for this problem, one is an in-protocol solution and the other is an out-of-protocol solution. In this version of draft, we describe both of them for further discussion.

3.1. Address mapping with session id

This is an out-of-protocol solution. The basic idea is that the concentrator(server) inspects the incoming DHCP packets, keeps track of the mapping between the DHCP session id and the I-IP address of the packet. When sending out a DHCP packet, the concentrator will use the session id in the packet to look up corresponding I-IP address for encapsulation. Here the session id could be any field in the DHCP packet that can be used to distinguish different clients, such as MAC address, transaction-id, etc. The mapping need to last for only the lifetime of two-time handshake.

Figure 2 provides an example using MAC as session id. When receiving a DHCPDISCOVER message, the concentrator stores the mapping between the MAC address and I-IP address in encapsulation header. Then the concentrator decapsulates the packet and hands the packet to upper layer. When the upper layer passes down the corresponding DHCPACK packet, the concentrator will look up the I-IP address in the mapping table, using the MAC address in the DHCPACK packet. This I-IP address will be used as encapsulation destination address. Then the mapping can expire. Similar procedure happens when the concentrator receives a DHCPREQUEST and sends out a DHCPACK.

This method is transparent to the DHCP process. There’s no protocol extension required. However, the concentrator do need to inspect every encapsulated packet to filter out DHCP packets.

<table>
<thead>
<tr>
<th>DHCP EVENT</th>
<th>initiator</th>
<th>concentrator</th>
<th>BEHAVIOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>allocating a new</td>
<td>---DHCPDISCOVER--&gt;</td>
<td>store I-IP-MAC mapping</td>
<td></td>
</tr>
<tr>
<td>network address</td>
<td>&lt;-----DHCPOFFER----&gt;</td>
<td>look up I-IP using MAC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;-----DHCPREQUEST----&gt;</td>
<td>mapping expires</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;-----DHCPACK-----</td>
<td>store I-IP-MAC mapping</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>look up I-IP using MAC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>mapping expires</td>
<td></td>
</tr>
<tr>
<td>address renewal</td>
<td>---DHCPREQUEST----&gt;</td>
<td>store IPv6-MAC mapping</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;-----DHCPACK-----</td>
<td>look up I-IP using MAC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>mapping expires</td>
<td></td>
</tr>
</tbody>
</table>
3.2. Leveraging relay agent option

Unlike the first solution, the second solution is an in-protocol solution. We can see that what’s actually needed in this problem is an I-IP encapsulation address for every DHCP packet. We can have the DHCP client to include this information in every DHCP packet it sends out. This document suggests to use the Agent Circuit ID Sub-option in DHCP Relay Agent Information Option (Option 82) [RFC4925] to carry the I-IP address information.

Having the client doing this, the operations on the concentrator can be significantly simplified. The receiving and decapsulating procedure of the DHCP packet can be identical to regular data packet. The DHCP server process will not modify Option 82 in a DHCP packet, and this option will be included in the DHCP reply packet. When the upper layer passes down the DHCP reply packet, the concentrator will look into the packet and find the encapsulation address in Option 82. Then the encapsulation can be done easily.

This method doesn’t need per-packet inspecting when decapsulating packet, and doesn’t need address mapping maintenance, either. However, it’s a "misuse" of Option 82 in some level, since there’s actually no DHCP relay involved. Another possibility is that we can define a new DHCP option for this specific usage if it’s necessary.
4. Acknowledgement

The authors would like to thank Alain Durand, Yiu L. Lee and Ted Lemmon for their valuable comments on this draft.
5. References

5.1. Normative References


5.2. Informative References

Authors’ Addresses

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

Peng Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing  100084
P.R.China

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

Jianping Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing  100084
P.R.China

Phone: +86-10-6278-5983
Email: jianping@cernet.edu.cn