Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 12, 2012

Y. Cui
P. Wu
J. Wu
Tsinghua University
July 11, 2011

DHCPv4 Behavior over IP-IP tunnel
draft-cui-softwire-dhcp-over-tunnel-01

Abstract

This document analyzes the scenario in which DHCPv4 interaction 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 are both in-protocol 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 packets.

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 12, 2012.

Copyright Notice

Table of Contents

1. Introduction

The DHC protocol[RFC2131] wasn't designed with tunnel environment considerations. However, due to the development of tunnel-based mechanisms, the demand to apply DHCP in tunnel environment arises, especially in the context of IPv6 transition. 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 hosts, across a heterogenous network. If the non-native IP addresses of the clients are provided by the concentrator side, this address provisioning needs to cross the heterogeneous network, too.

One transition mechanism that requires DHCP over tunnel is documented in [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 document suggests to achieve this by tunneling DHCPv4 between the 4over6 initiator(DHCPv4 client) and 4over6 concentrator(DHCPv4 server).

Two main flavours of solutions may be considered:

- Use DHCPv6 to provision IPv4-related connectivity, since IPv4 address can be embedded into IPv6 address field. To achieve this mode, dedicated options are needed to convey IPv4-related information, such as IPv4 address of DNS server, NTP server, etc.

- Use DHCPv4 and sustain it in the tunnel environment. Unlike the previous approach where only DHCPv6 is used for both IPv4 and IPv6 connectivity, this approach consists in maintaining the separation between IPv4 and IPv6 connectivity information. It allows to maintain the IPv4 service without requiring major modification of IPv6-related provisioning resources, and perserves DHCP as an IPv4-related information carrier. This document focuses on this flavour.

2. Teminology

This document makes use of the following terms:

- DHCPv4 refers to IPv4 DHCP [RFC2131].
- DHCPv4 client (or client) denotes a node that initiates requests to obtain configuration parameters from one or more DHCP servers [RFC2131].
- DHCPv4 server (or server) refers to a node that responds to requests from DHCP clients [RFC2131].

3. 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. DHCP DISCOVER and DHCP REQUEST packets cannot reach the other end since they are broadcast packets; DHCP OFFER and DHCP ACK/NAK packets cannot reach the other end either, when they are broadcast packets or unicast packets forwarded by MAC address. Therefore a tunnel between the client and server is required to build a virtual link. Besides, when the middle network is IPv6-only, all DHCPv4 packets can not go through the network.

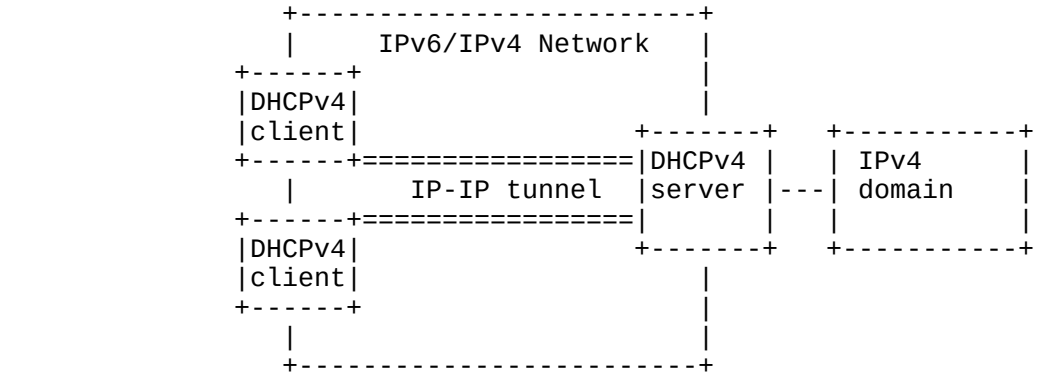

Figure 1 Scenario of DHCPv4 over tunnel

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 a 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 will

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 DHCP packets can be broadcast address. So we need some extra effort to make the encapsulation of DHCP packets work, i.e., make the concentrator encapsulate each DHCP packet with the I-IP address of the right initiator and hence send it to the right initiator.

4. In-protocol and Out-of-protocol Solutions

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.

4.1. 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 needs 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 DHCPOFFER packet, the concentrator will look up the I-IP address in the mapping table, using the MAC address in the DHCPOFFER 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 need to inspect every encapsulated packet to filter out DHCP packets.

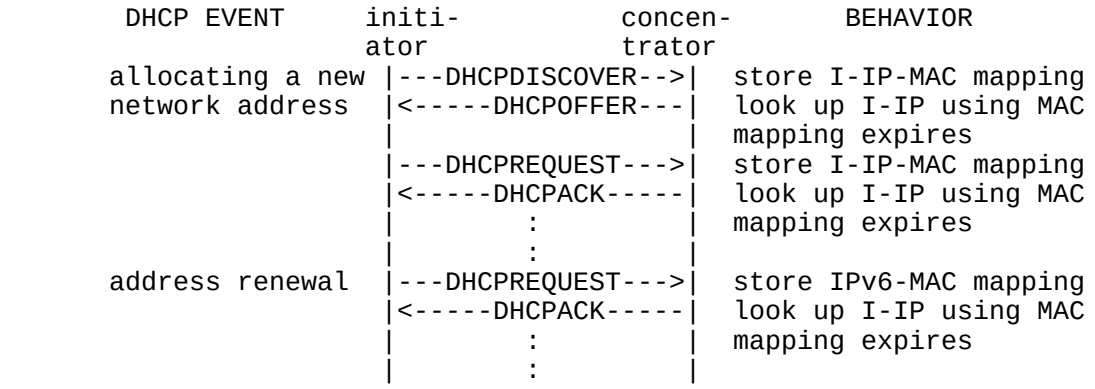

Figure 2 4over6 concentrator: DHCP behavior

4.2. Leveraging Relay Agent Option

Unlike the first solution, the second solution is an in-protocol solution. We can see that what is actually needed to solve 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) [RFC3046] 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 is necessary.

5. Acknowledgement

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

6. References

6.1. Normative References

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

[RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.

[RFC4925] Li, X., Dawkins, S., Ward, D., and A. Durand, "Softwire Problem Statement", RFC 4925, July 2007.

6.2. Informative References

[I-D.boucadair-dhcpv6-shared-address-option] Boucadair, M., Levis, P., Grimault, J., Savolainen, T., and G. Bajko, "Dynamic Host Configuration Protocol (DHCPv6) Options for Shared IP Addresses Solutions", draft-boucadair-dhcpv6-shared-address-option-01 (work in progress), December 2009.

[I-D.cui-softwire-host-4over6] Cui, Y., Wu, J., Wu, P., Metz, C., Vautrin, O., and Y. Lee, "Public IPv4 over Access IPv6 Network", draft-cui-softwire-host-4over6-06 (work in progress), July 2011.

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