Configuring IPv4 over IPv6 Networks

Transitioning with DHCP

Yong Cui • *Tsinghua University, China*

Qi Sun • *Beijing University of Posts and Telecommunications*

Ke Xu • *Tsinghua University, China*

Wendong Wang • *Beijing University of Posts and Telecommunications*

Ted Lemon • *Nominum*

To transition to IPv6, ISPs must determine how to configure IPv4 over IPv6 networks using the Dynamic Host Configuration Protocol (DHCP). However, multifarious requirements result in various solutions. The authors sort out the requirements for configuring an IPv4 node in IPv6 networks, survey IETF standardization efforts, and give recommendations on this topic.

IPv4 address exhaustion is forcing the Internet to transition to IPv6.[1] Since 2000, the IETF has spent considerable effort on standardizing IPv6 transition techniques, including 6over4,[2] Stateless IP/ICMP Translation (SIIT),[3] Network Address Translation-Protocol Translation (NAT-PT),[4] and 6to4.[5] However, none of these techniques were deployed at a large scale, and some were even deprecated.[6] A key issue is that they failed to manage heterogeneous addresses effectively, leading to problems such as network planning, routing scalability, and so on. Another challenge is that operators are facing situations in which IPv6-only access networks are deployed, but the majority of Internet services remain in IPv4.

Configuring IPv4 over heterogeneous IPv6 can preserve service continuity and promote the transition to IPv6. The Dynamic Host Configuration Protocol (DHCP) is preferred for this configuration, but DHCPv4[7] and DHCPv6[8] aren't interoperable. DHCPv6 can't configure IPv4, although it works in IPv6. DHCPv4 is designed to provision IPv4 resources, but it breaks when the transmission networks are IPv6-only.

The IETF has proposed a series of solutions since 2009, and various working groups — including DHC, Softwires, and Sunset4 — are working to resolve the issue. Different solutions would not only considerably affect the design and deployment of IPv6 transition techniques but would also influence the IPv6 world in the long run. The IETF community is trying to avoid barriers on future IPv6 development while still leveraging DHCPv4 investments during the IPv6 transition period.

Here, we classify the proposed solutions into three categories: DHCPv6-based mechanisms, DHCPv4-based mechanisms, and a mechanism combining DHCPv4 and DHCPv6. We analyze the requirements and introduce the various solutions according to their classification. This lets us provide recommendations on selection.

Network Architecture and Requirements

The IPv6 network scale has been expanding over the years. However, plenty of Internet services still remain in IPv4, resulting in users preferring IPv4 for better experience. To continue IPv4 services while promoting IPv6 deployment, ISPs are

1089-7801/14/$31.00 © 2014 IEEE

focusing on the IPv4-over-IPv6 transition scenario.

Figure 1 illustrates the architecture of the IPv4-over-IPv6 transition. The access network between hosts and customer premises equipment (CPE), and border routers (BR) is IPv6-only. Dual-stack border routers connect to IPv6 networks and the IPv4 Internet. An IPv4-in-IPv6 tunnel sustains IPv4 data traffic. To provide IPv4 services and keep end-to-end transparency, operators should allocate global IPv4 addresses across IPv6 networks. Distributing IPv4 addresses also benefits network management compared to introducing carrier-grade NAT (CGN).

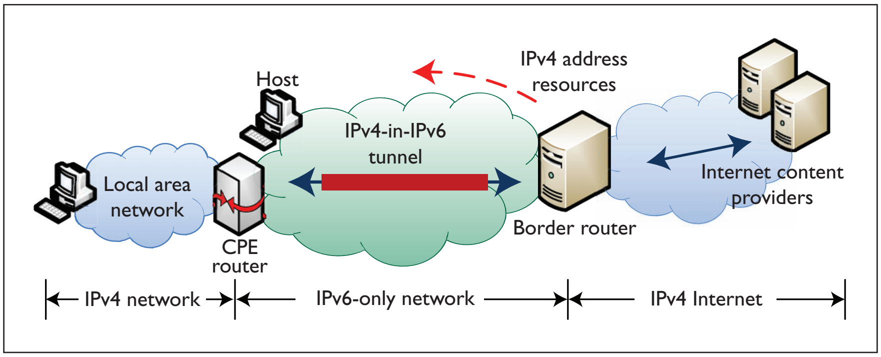

Figure 1. IPv4-over-IPv6 transition architecture. Public IPv4 address resources are allocated from the ISP side to the user-end to set up IPv4-in-IPv6 tunnels. IPv4 traffic traverses IPv6-only networks through tunnels, and the ISP avoids having to maintain carrier-grade NAT.

Within this architecture, different levels of requirements apply to IPv4 node configuration. The basic information is the IPv4 address or shared IPv4 address (that is, an address with a restricted layer-4 port set), which an IPv4 node can get through a predetermined (static) mapping with an IPv6 address/prefix. If the operator's IPv4 address spaces are limited and scattered, dynamic IPv4 leasing is needed to fully utilize scarce IPv4 addresses. Decoupled IPv4 and IPv6 addressing schemes can simplify network planning. To provide some services, ISPs might demand other IPv4 configuration parameters such as Network Time Protocol and Session Initiation Protocol server addresses.

DHCP is designed to automatically allocate network addresses as well as other service configuration parameters to end-users. However, the incompatibility between DHCPv4 and DHCPv6 requires that we develop extensions to fulfill the aforementioned requirements. Let's look at the three categories of solutions.

DHCPv6-Based Solutions

An ISP is likely to run DHCPv6 only in an IPv6 network. Extending DHCPv6 to configure IPv4 can leverage DHCPv6 infrastructures: one server would be adequate to configure both IPv4 and IPv6. DHCPv6-based solutions have the advantage of using DHCPv6 options. IPv4 address resources are simply put into DHCPv6 options and conveyed to clients along with IPv6 configuration processes. The DHCPv6 server manages IPv4 address resources statically or dynamically, and the client is able to configure the IPv4 stack.

The Softwire DHCPv6 Options mechanism aims to provision only basic IPv4 information[9] – that is, IPv4 prefixes, addresses, or addresses with a layer-4 port-set. It enables operators to treat IPv4 address information as a DHCPv6 configuration parameter and provision it statelessly through the DHCPv6 option. So, an IPv4 address's lifetime is tied to that of an IPv6 address, as with other DHCPv6 parameters. The server must predetermine the mapping relationship between IPv4 addresses and IPv6 addresses or prefixes, which puts additional requirements on address planning.

By adding a specific lifetime to DHCPv6 options, DHCPv6 Shared Address Options dynamically manages IPv4 address information.[10] IPv4 addresses can be reclaimed for future allocation once they expire. However, maintaining dynamic IPv4 leasing would require dramatic modifications to the DHCPv6 server, which is more compliant with DHCPv4 than DHCPv6.

In addition to IPv4 addresses, DHCPv6 can deliver other IPv4 configuration parameters. One possibility is to redefine the required DHCPv4 options in a "private" DHCPv6 option space. ISPs must determine which DHCPv4 options they might need in the future. However, nobody can guarantee a convincing list. Another possibility is to use a DHCPv6 Container Option for all DHCPv4 options.[11] Such attempts would require rework on DHCPv6 clients and servers when importing every single DHCPv4 option. Furthermore, the imported options would "pollute" the DHCPv6 option space permanently, regardless of whether IPv4 were still in use.

DHCPv6+DHCPv4-over-Softwire uses DHCPv6 options to statically allocate IPv4 addresses, with which the client sets up IPv4-over-IPv6 softwires.[12] The client requests additional IPv4 configuration information from a standalone DHCPv4 server through the softwire concentrator, which performs the encapsulation and decapsulation functions. The client must know the DHCPv4 server's IPv4 address and the softwire concentrator's IPv6 address in advance. The IPv4 address is used to identify the DHCPv4 server, and the IPv6 address helps establish softwires.

DHCPv4-Based Solutions

DHCPv4 is feasible for configuring IPv4, provided that it can survive in IPv6 networks. DHCPv4-based solutions use transport patterns other than

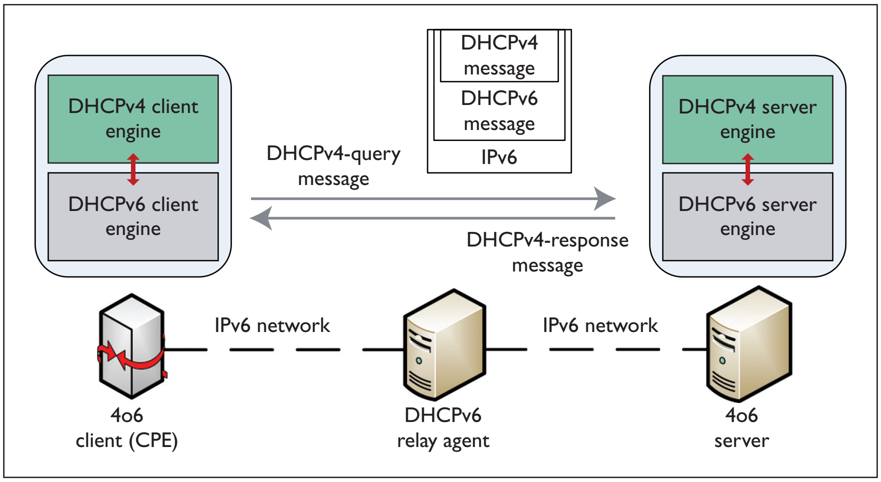

Figure 2. DHCPv4 over DHCPv6 transport. DHCPv4 and DHCPv6 engines are located on the same DHCP server or client. Two new DHCPv6 messages are specialized to encapsulate DHCPv4 messages for transport over IPv6 networks. Both IPv6 multicast and unicast are viable for communication.

IPv4, such as DHCPv4 over IP-IP tunnel or DHCPv4 over IPv6. Thus, ISPs can allocate IPv4 address resources across the IPv6-only environment while preserving DHCPv4 features. Either way keeps the DHCPv4 engine almost intact. Using IPv6 unicast places an additional requirement on the client to learn the server's IPv6 address beforehand.

DHCPv4 over Tunnel

DHCPv4 over Tunnel[13,14] builds DHCPv4 functionalities on IPv4-in-IPv6 tunnels. The client and server, as tunnel endpoints, encapsulate a DHCPv4 packet (with an IPv4 header) into IPv6 before sending it out. On receiving tunnel packets, the client and server sniff DHCPv4 packets and decapsulate them for further processing. To correctly reply to multiple clients, the server must determine their IPv6 addresses. It can maintain the mapping between the client ID and source IPv6 address. Also, the client can extend the Agent Circuit ID Suboption in the DHCPv4 Relay Agent Option to record its source IPv6 address. However, this method misuses the Relay Agent Option, given that this process doesn't actually involve a relay agent.

However, the mechanism causes the classic chicken-and-egg problem. DHCPv4 must depend on the IPv4-in-IPv6 tunnel to assign an IPv4 address, but no IPv4 configuration can be used to set up the tunnel. Another issue is that this mechanism restricts DHCPv4 transportation to tunneling, which requires significant modifications to DHCPv4 architecture because the client works on IPv4 broadcast rather than tunnels.

DHCPv4 over IPv6

Instead of using tunneling, DHCPv4 over IPv6 takes DHCPv4 messages as IPv6 UDP payload.[15] To avoid modifying the DHCPv4 client, this approach adds a *client relay agent* (CRA) function to the client side to adapt DHCPv4 packets for IPv6 transport. The server is IPv6-aware — that is, it extracts DHCPv4 messages from IPv6 packets and handles them as regular DHCPv4 servers do.

To reuse the unmodified DHCPv4-only server, this mechanism extends the DHCPv4 relay agent to deal with DHCPv4-over-IPv6 traffic. The CRA6ADDR suboption, a Relay Agent Suboption, is defined to record the inbound request's source IPv6 address. The relay agent adds this suboption when relaying messages to the server and extracts the IPv6 address when responding to the client. The interactions between the extended relay agent and the unmodified server are the same as in a regular DHCPv4 environment.

DHCPv4 over IPv6 retains DHCPv4 features and guarantees that IPv4 and IPv6 address management remain separate. An ISP can avoid upgrading an existing DHCPv4-only server by deploying an extended relay agent. The changes to servers and relay agents can be abandoned with IPv4 once the IPv6 transition completes. However, this mechanism is more complex compared to DHCPv6-based solutions. It also requires deploying DHCPv4 servers or relay agents at the border of an IPv6-only network, when ISPs might want to avoid investing in DHCPv4 infrastructures. Moreover, it isn't compatible with a DHCPv6-only network.

DHCPv4 over DHCPv6: An Architectural Solution

Configuring IPv4 over IPv6 relates to not only IPv4 address allocation but also future DHC use cases. DHCPv6- and DHCPv4-based mechanisms are more like workarounds. An architectural solution would require breaking the isolation between DHCPv4 and DHCPv6, so that ISPs could reuse the former without damaging the latter. Most current commercial DHCP servers support both protocols in one device, making it possible to integrate the two.

The IETF has proposed DHCPv4 over DHCPv6 for this purpose (see Figure 2).[16] The fundamental principle is to convey DHCPv4 messages within DHCPv6 messages. The DHCP 4o6 server and client integrate both DHCPv4 and DHCPv6 engines. The DHCPv4 engine works as the core for processing DHCPv4 messages, whereas the DHCPv6 engine transports these messages over IPv6-only networks. Two new DHCPv6 messages are specialized for the communications: DHCPv4-query and DHCPv4-response, along

Table 1. Comparisons of IPv4 over IPv6 configuration solutions.

Solution	Supports dynamic IPv4 leasing	Supports existing DHCPv4 options	Separate v4 and v6 address allocation	Avoids polluting DHCPv6 options	Compatible with DHCPv6	Complexity of updating network facilities
Shared IPv4 Address Options	Yes, but difficult	No	No	No	Yes	High
Softwire DHCPv6 Options	No	No	No	No	Yes	Low
DHCPv6 Option Container for v4 options	N/A	Yes	N/A	No	Yes	High
DHCPv6+DHCPv4 over Softwire	No	Yes	No	Yes	Yes	High
DHCPv4 over tunnel	Yes	Yes	Yes	Yes	No	Very high
DHCPv4 over IPv6	Yes	Yes	Yes	Yes	No	High
DHCPv4 over DHCPv6	Yes	Yes	Yes	Yes	Yes	High

with the DHCPv4 Message option for encapsulating DHCPv4 messages. The DHCPv4-query and DHCPv4-response messages convey the client-to-server and server-to-client DHCPv4 messages, respectively. The two messages specify a flag-bits field to provide additional information for the client and server. The unicast flag is the first bit of the field in the DHCPv4-query message, which indicates to the server whether the inner DHCPv4 message should have been sent to IPv4 unicast or broadcast. The information is valuable for the server to determine the client's state.

We define the 4o6 Server Address option as the trigger for DHCPv4 over DHCPv6 to avoid collisions with future DHCPv6 deployment. If the option contains no IPv6 address, the client uses the well-known `All_DHCP_Relay_Agents_and_Servers` multicast address[8] as the destination. Otherwise, the client just sends requests to the IPv6 addresses listed in the option. With this mechanism, communicating through IPv6 multicast and unicast is viable.

Although a regular DHCPv6 relay agent doesn't recognize the new messages, it can simply wrap the received messages into a relay-forward message or extract the content from a relay-reply message, and then perform forwarding.[17] This enables DHCPv6 relay agents to accommodate future innovations.

DHCPv4 over DHCPv6 takes DHCPv6 as the foundation for communications. All DHCPv4 features are preserved in terms of dynamic IPv4 leasing management, other service parameter configurations, DHCPv4 failover, leasequery, and so on. ISPs and vendors can also leverage operational experience in DHCPv4 system and server/client codes. The IPv4 and IPv6 configuration processes are separate from each other. Compared to the topology of DHCPv4-based solutions, DHCPv4 over DHCPv6 is compatible with DHCPv6 architecture without introducing standalone DHCPv4 servers at the borders. The cost is relatively high complexity to implement new DHCPv6 messages. However, the CPEs, hosts, and BRs on which DHCP functions run are bound to update for transitioning to IPv6. So, implementation complexity doesn't matter that much.

Implementations and Comparisons

Several IETF working groups, including DHC, Softwire, and Sunset4, are engaged in discussions about configuring IPv4 over IPv6 through DHCP. Furthermore, a working draft analyzes the recent proposals.[12] Some vendors, ISPs, and institutes have demonstrated the proposed mechanisms by testing or demoing running codes. To our knowledge, Huawei, ISC, and Tsinghua University (THU) have implementations of DHCPv4 over IPv6.[18] THU has implemented a prototype of DHCPv4 over DHCPv6, which is merging with ISC's new DHCP project (http://bind10.isc.org/wiki/Kea).

Nevertheless, keeping the balance between a solution's feasibility for the time being and the need to fulfill future demands is difficult. Table 1 compares the proposed mechanisms for configuring IPv4 over IPv6 networks.

When choosing among different solutions, ISPs must consider their current network investments and the potential requirements for IPv6 transition. If the ISP only needs IPv4 address information for statelessly setting up softwires, Softwire DHCPv6 Options is suitable. For ISPs with scattered IPv4 address spaces, DHCPv4 over DHCPv6 fits well through dynamic IPv4 leasing. If additional IPv4 configuration information is required, DHCPv4 over DHCPv6 is the best choice.

Currently, Softwire DHCPv6 Options and DHPCv4 over DHCPv6 are approved in the Softwire and DHC working groups, respectively. Softwire Unified

CPE[19] intends to combine the two for provisioning a CPE device. These solutions promote IPv6 adoption while preserving IPv4 investment during the IPv6 transition period.

Acknowledgments

This work is supported by National Natural Science Foundation of China project nos. 61120106008 and 61161140454, National 863 project no. 2013AA010401, China Next Generation Internet project CNGI-12-03-003, and New Generation Broadband Wireless Mobile Communication Network of the National Science and Technology Major project no. 2012ZX03005001.

References

1. P. Wu et al., "Transition from IPv4 to IPv6: A State-of-the-Art Survey," *IEEE Comm. Surveys & Tutorials*, Dec. 2012, pp. 1407–1424.
2. B. Carpenter et al., *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels*, IETF RFC 2529, Mar. 1999; www.ietf.org/rfc/rfc2529.txt.
3. E. Nordmark, *Stateless IP/ICMP Translation Algorithm (SIIT)*, IETF RFC 2765, Feb. 2000; www.ietf.org/rfc/rfc2765.txt.
4. G. Tsirtsis et al., *Network Address Translation-Protocol Translation (NAT-PT)*, IETF RFC 2766, Feb. 2000; www.ietf.org/rfc/rfc2766.txt.
5. B. Carpenter et al, *Connection of IPv6 Domains via IPv4 Clouds*, IETF RFC 3056, Feb. 2001; www.ietf.org/rfc/rfc3056.txt.
6. C. Aoun et al., *Reasons to Move the Network Address Translator-Protocol Translator (NAT-PT) to Historic Status*, IETF RFC 4966, July 2007; www.ietf.org/rfc/rfc4966.txt.
7. R. Droms, *Dynamic Host Configuration Protocol*, IETF RFC 2131, Mar. 1997; www.ietf.org/rfc/rfc2131.txt.
8. R. Droms et al., *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, IETF RFC 3315, July 2003; www.ietf.org/rfc/rfc3315.txt.
9. T. Mrugalski et al., "DHCPv6 Options for Configuration of Softwire Address and Port Mapped Clients," IETF Internet draft, work in progress, Mar. 2014.
10. M. Boucadair et al., "Dynamic Host Configuration Protocol (DHCPv6) Options for Shared IP Addresses Solutions," IETF Internet draft, work in progress, Dec. 2009.
11. Q. Sun and Y. Cui, "DHCPv6 Option for IPv4 Configuration," IETF Internet draft, work in progress, Feb. 2013.
12. B. Rajtar et al., "Provisioning IPv4 Configuration over IPv6 Only Networks," IETF Internet draft, work in progress, Feb. 2014.
13. Y. Cui et al., "DHCPv4 Behavior over IP-IP Tunnel," IETF Internet draft, work in progress, July 2011.
14. O. Troan, "DHCPv4 over A+P Softwires," IETF Internet draft, work in progress, June 2013.
15. Y. Cui et al., "DHCPv4 over IPv6 Transport," IETF Internet draft, work in progress, Oct. 2013.
16. Q. Sun et al., "DHCPv4 over DHCPv6 Transport," IETF Internet draft, work in progress, Feb. 2014.
17. Y. Cui, Q. Sun, and T. Lemon, "Handling Unknown DHCPv6 Messages," IETF Internet draft, work in progress, Mar. 2014.
18. Y. Chen et al., "Lightweight 4over6 Interop Report," IETF Internet draft, work in progress, Nov. 2012.
19. M. Boucadair et al., "Unified IPv4-in-IPv6 Softwire CPE," IETF Internet draft, work in progress, May 2013.

Yong Cui is a full professor at Tsinghua University, China. His research interests include computer network architecture and mobile computing. Cui has a PhD in computer science from Tsinghua University. He has published three IETF RFCs on IPv6 transition technologies, and he cochairs the IETF Softwire Working Group, which focuses on tunneling technology for IPv6 transition. Contact him at cy@csnet1.cs.tsinghua.edu.cn.

Qi Sun is a master student at the Beijing University of Posts and Telecommunications (BUPT). His research interests include IPv4-IPv6 transition and DHCP. Sun has a BS in communication engineering from BUPT. He has coauthored nine IETF drafts on IPv6 transition techniques. Contact him at sunqifs@bupt.edu.cn.

Ke Xu is a full professor at Tsinghua University, China. His research interests include next-generation Internet and peer-to-peer and overlay networks. Xu has a PhD in computer science from Tsinghua University. He is a member of ACM. Contact him at xuke@tsinghua.edu.cn

Wendong Wang is a full professor at Beijing University of Posts and Telecommunications (BUPT). His research interests include future Internet architecture, SDN technologies, and network or service QoS. Wang received an ME in computer science from BUPT. He's contributed to three IETF drafts related to the Dynamic Host Configuration Protocol. Contact him at wdwang@bupt.edu.cn.

Ted Lemon is one of the founders of Nominum, the Internet Area Director at the IETF, and, until recently, a cochair of the DHC Working Group. His research interests include IPv6, DHCP, and DNS. Lemon has published 11 RFCs. Contact him at ted.lemon@nominum.com.