Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

Y. Cui
J. Wu
P. Wu
Tsinghua University
C. Metz
Cisco Systems, Inc.
O. Vautrin
Juniper Networks
Y. Lee
Comcast

March 14, 2011

Public IPv4 over Access IPv6 Network
draft-cui-softwire-host-4over6-04

Abstract

This draft proposes a mechanism for bidirectional IPv4 communication between IPv4 Internet and end hosts or IPv4 networks sited in IPv6 access network. This mechanism follows the softwire hub & spoke model and uses IPv4-over-IPv6 tunnel as basic method to traverse IPv6 network. By allocating public IPv4 addresses to end hosts/networks in IPv6, it can achieve IPv4 end-to-end bidirectional communication between these hosts/networks and IPv4 Internet. This mechanism is an IPv4 access method for hosts and IPv4 networks sited in IPv6.

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 September 15, 2011.

Copyright Notice

Table of Contents

1. Introduction

Global IPv4 addresses are running out fast. Meanwhile, the demand for IP address is still growing and may even burst in potential circumstances like "Internet of Things". To satisfy the end users, operators have to push IPv6 to the front, by building IPv6 networks and providing IPv6 services.

When IPv6-only network are widely deployed, users of those networks will probably still need IPv4 connectivity. This is because part of Internet will stay IPv4-only for a long time, and network users in IPv6-only network will communicate with network users sited in the IPv4-only part of Internet. This need could eventually decrease with the general IPv6 adoption.

Network operators should provide IPv4 services to IPv6 users to satisfy their needs, usually through tunnels. This type of IPv4 services differ in provisioned IPv4 addresses. If the users can't get public IPv4 addresses (e.g., new network users join an ISP which don't have enough unused IPv4 addresses), they have to use private IPv4 addresses on the client side, and IPv4-private-to-public translation is required on the carrier side, as is described in Dual-stack Lite[I-D.ietf-softwire-dual-stack-lite]. Otherwise the users can get public IPv4 addresses, and use them for IPv4 communication. In this case, translation on the carrier side won't be necessary. The network users and operators can avoid all the issues raised by translation, such as ALG, NAT traversal, state maintenance, etc. Note that this "public IPv4" situation is actually quite common. There're approximatively 2^32 network users who are using or can potentially get public IPv4 addresses. Most of them will switch to IPv6 sooner or later, and will require IPv4 services for a significant period after the switching. This draft focuses on this situation, i.e., to provide IPv4 access for users in IPv6 networks, where public IPv4 addresses are still available for allocation.

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

Public 4over6: Public 4over6 is the mechanism proposed by this draft. Generally, Public 4over6 supports bidirectional communication between IPv4 Internet and IPv4 hosts or local networks in IPv6 access network, by leveraging IPv4-in-IPv6 tunnel and public IPv4 address allocation.

4over6 initiator: in Public 4over6 mechanism, 4over6 initiator is the IPv4-in-IPv6 tunnel initiator located on the user side of IPv6 network. The 4over6 initiator can be either a dual-stack capable host or a dual-stack CPE device. In the former case, the host has both IPv4 and IPv6 stack but is provisioned with IPv6 access only. In the latter case, the CPE has both IPv6 interface for access to ISP network and IPv4 interface for local network connection; hosts in the local network can be IPv4-only.

4over6 concentrator: in Public 4over6 mechanism, 4over6 concentrator is the IPv4-in-IPv6 tunnel concentrator located in IPv6 ISP network. It’s a dual-stack router which connects to both the IPv6 network and IPv4 Internet.

4. Deployment scenario

4.1. Scenario description

The general scenario of Public 4over6 is shown in Figure 1. Users in an IPv6 network take IPv6 as their native service. Some users are end hosts which face the ISP network directly, while others are local networks behind CPEs, such as a home LAN, an enterprise network, etc. The ISP network is IPv6-only rather than dual-stack, which means that ISP can't provide native IPv4 access to its users; however, it's acceptable that one or more routers on the carrier side become dual-stack and get connected to IPv4 Internet. So if network users want to connect to IPv4, these dual-stack routers will be their "entrances".

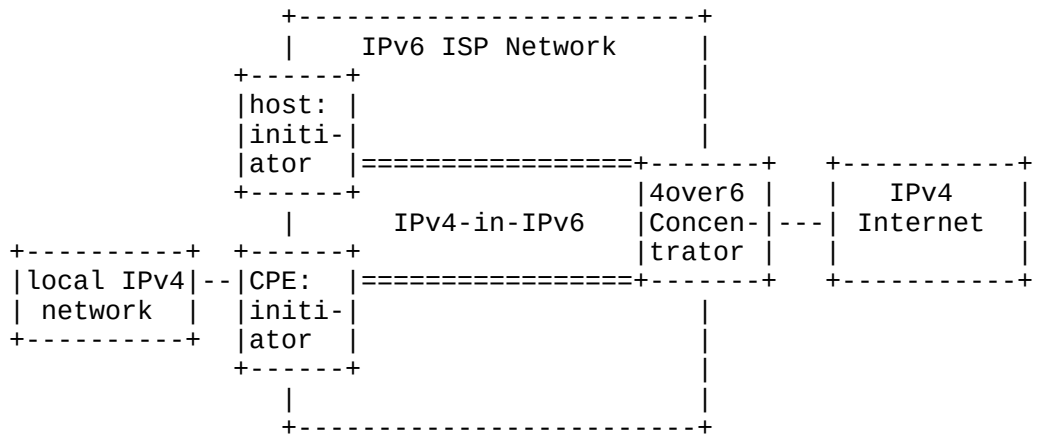

Figure 1 Public 4over6 scenario

4.2. Communication requirements

Before getting into any technical details, the communication requirements should be stated. The first one is that, 4over6 users require IPv4-to-IPv4 communication with the IPv4 Internet. An IPv4 access service is needed rather than an IPv6-to-IPv4 translation service. (IPv6-to-IPv4 communication is out of the scope of this draft.)

Second, 4over6 users require public IPv4 addresses rather than private addresses. Public IPv4 address means there's no IPv4 CGN along the path, so the acquired IPv4 service is better. In particular, some hosts may be application servers, public address

works better for reasons like straightforward access, direct DNS registration, no stateful mapping maintenance on CGN, etc. For the direct-connected host case, each host should get one public IPv4 address. For the local IPv4 network case, there're actually two subcases: one is that every CPE gets one public IPv4 address while local networks remains private IPv4, the other is that end hosts in local networks get public IPv4 addresses. In the first subcase, though the CPE has to run an IPv4 NAT, it's still much better than the situation that involves a CGN, since this NAT is in local network and can be configured and managed by the users.

Third, translation is not preferred in this scenario. If this IPv4-to-IPv4 communication is achieved by IPv4-IPv6 translation, it'll needs double translation along the path, one from IPv4 to IPv6 and the other from IPv6 back to IPv4. It's quite complicated. Contrarily a tunnel can achieve the IPv4-over-IPv6 traversing easily. That's the reason this draft follows the hub & spoke softwire model.

5. Public 4over6 Mechanism

5.1. Address allocation

Public 4over6 can be generally considered as IPv4-over-IPv6 hub & spoke tunnel using public IPv4 address. Each 4over6 initiator will use public IPv4 address for IPv4-over-IPv6 communication. As is described above, in the host initiator case, every host will get one IPv4 address; in the NATed CPE case, every CPE will get one IPv4 address, which will be shared by hosts behind the CPE; in the non-NAT CPE case, every host behind the CPE will get one IPv4 address.

The key problem here is IPv4 address allocation over IPv6 network, from ISP device(s) to separated 4over6 initiators. Native IPv4 address allocation is done either in a dynamic way throug DHCPv4, or in a static way through manual configuration. Public 4over6 should support both. DHCPv4 over IPv6 can be achieved upon IPv4-in-IPv6 tunnel between ISP device and 4over6 initiators. As to manual configuration, 4over6 users and the ISP operators should negotiate beforehand to authorize the IPv4 address. In addition, in the non-NAT CPE case, the address allocation should pass through the CPE initiator and reach IPv4 hosts. This will require a DHCP relay function on the CPE.

Along with this address allocation, the concentrator needs to maintain the address mappings between the allocated IPv4 address and IPv6 address of 4over6 initiators. This is required to provide correct destination address for encapsulation. There are several ways to maintain this mapping: DHCPv4-driven updating, traffic snooping and manual configuration. This draft recommends the first way since it naturally supports bidirectional communication. The next two subsection adopts the first method and describes it in detail. A comparison with traffic snooping is given in section 5.4.

5.2. 4over6 concentrator behavior

4over6 concentrator represents the IPv4-IPv6 border router working as the remote tunnel endpoint for 4over6 initiators, with its IPv6 interface connected to the IPv6 network, IPv4 interface connected to the IPv4 Internet, and a tunnel interface supporting IPv4-in-IPv6 encapsulation and decapsulation. There’s no CGN on the 4over6 concentrator, it won’t perform any translation function; instead, 4over6 concentrator maintains an IPv4-IPv6 address mapping table for IPv4 data encapsulation.

4over6 concentrator is responsible for IPv4 address allocation to 4over6 initiators. For static allocation, the concentrator just install the IPv4-IPv6 address mapping into the mapping table after

negotiating with a 4over6 user, and delete the mapping when the user doesn't need 4over6 anymore. As to dynamic allocation, the concentrator should either run a DHCPv4 server on the tunnel interface to dynamically allocate public addresses to 4over6 initiators, or perform the DHCPv4 relay functions and leave the actual address allocation job to a dedicated DHCPv4 server located in IPv4. In both cases, when allocating an address, the concentrator should install an entry of the allocated IPv4 address and the initiator's IPv6 address into the address mapping table. This entry should be deleted when receiving a DHCP release or reaching a lease expiration of that IPv4 address. All these mapping updates are triggered by the DHCP process(see Figure 2). Note that in the DHCP relay case, the relay should be extended to maintain the lifetime of address leases.

The concentrator sends and receives DHCP packets using IPv4-in-IPv6 tunnel. The difficulty here is that before DHCP address allocation is done, the initiator may not have an IPv4 address, and the concentrator doesn't have an IPv4-IPv6 mapping for IPv4-in-IPv6 encapsulation of the DHCP packets. So when the concentrator receives an encapsulated DHCP packet from an initiator, it should temporarily store the mapping between its IPv6 source address and the MAC address in DHCP payload. This mapping will be used for encapsulation of outgoing DHCP packets. The concentrator should use the MAC address in the payload of an outing DHCP packet to match the correct IPv6 encapsulation destination address.

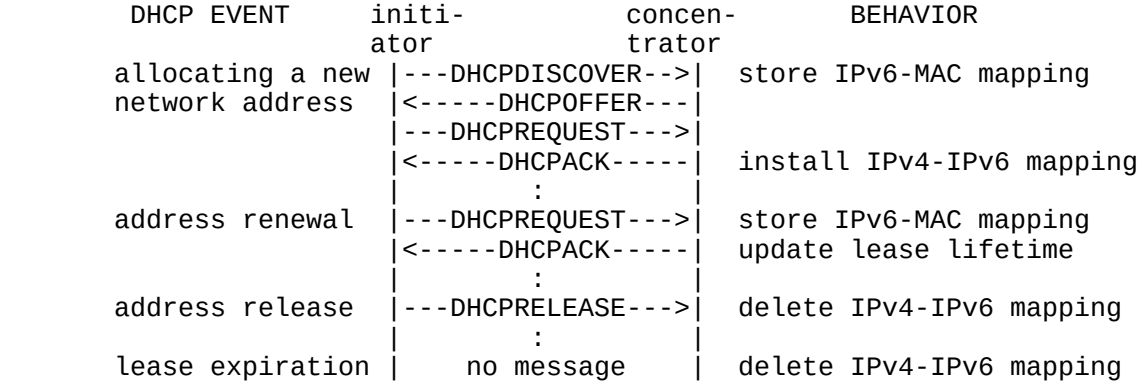

Figure 2 4over6 concentrator: DHCP behavior

On the IPv6 side, 4over6 concentrator decapsulates IPv4-in-IPv6 packets coming from 4over6 initiators. It removes the IPv6 header of every IPv4-in-IPv6 packet and forwards it to the IPv4 Internet. On the IPv4 side, the concentrator encapsulates the IPv4 packets

destined to 4over6 initiators. When performing the IPv4-in-IPv6 encapsulation, the concentrator uses its own IPv6 address as the IPv6 source address. As to the IPv6 destination address, the concentrator will look up the IPv4-IPv6 address mapping table, use the IPv4 destination address of the packet to find the correct IPv6 address. After the encapsulation, the concentrator sends the IPv6 packet on its IPv6 interface to reach an initiator.

The 4over6 concentrator, or its upstream router should advertise the IPv4 prefix which contains the IPv4 addresses of 4over6 users to the IPv4 side, in order to make these initiators reachable on IPv4 Internet.

Since the concentrator has to maintain the IPv4-IPv6 address mapping table, the concentrator is stateful in IP level. Note that this table will be much smaller than a CGN table, as there is no port information involved.

5.3. 4over6 initiator behavior

4over6 initiator has an IPv6 interface connected to the IPv6 ISP network, and a tunnel interface to support IPv4-in-IPv6 encapsulation. In CPE case, it has at least one IPv4 interface connected to IPv4 local network.

4over6 initiator should learn the 4over6 concentrator’s IPv6 address beforehand. For example, if the initiator gets its IPv6 address by DHCPv6, it can get the 4over6 concentrator’s IPv6 address through a DHCPv6 option[I-D.ietf-softwire-ds-lite-tunnel-option].

5.3.1. Host initiator

When the initiator is a direct-connected host, it’ll assign the allocated public IPv4 address to its tunnel interface. If the address allocation is static, the host should negotiate with the ISP operator beforehand. The host should learn the IPv4 address provisioned by the operator, and inform the operator its IPv6 address, to install the address mapping on the concentrator.

Usually, a host gets the public IPv4 address by DHCPv4 over an IPv4-in-IPv6 tunnel. A standard DHCPv4 client on the host will run on the tunnel interface to acquire IPv4 address. All the DHCPv4 packets generated by the client will be encapsulated and forwarded to the 4over6 concentrator, and all the DHCPv4 replies from the concentrator encapsulating in IPv6 will be decapsulated by the tunnel interface and handed to the DHCP client. This way the DHCP client can get a dynamic public IPv4 address from the concentrator, and assign it to the tunnel interface.

For IPv4 data traffic, the host performs the IPv4-in-IPv6 encapsulation and decapsulation on the tunnel interface, which has its IPv4 address already assigned. When sending out an IPv4 packet, it performs the encapsulation, using the IPv6 address of the 4over6 concentrator as the IPv6 destination address, and its own IPv6 address as the IPv6 source address. The encapsulated packet will be forwarded to the IPv6 network. The decapsulation on 4over6 initiator is simple. When receiving an IPv4-in-IPv6 packet, the initiator just drops the IPv6 header, and hands it to upper layer.

5.3.2. NATed CPE as initiator

The NATed CPE case is quite like the host initiator case. The IPv4 address allocation process between the CPE and the concentrator is the same with the corresponding process in host initiator case, and the allocated IPv4 address will be assigned to the tunnel interface of the CPE. The local IPv4 network won't take part in the public IPv4 allocation; instead end hosts will use private IPv4 addresses, possibly allocated by the CPE.

On data plan, the NATed CPE can be viewed as a regular IPv4 NAT(using tunnel interface as the NAT outside interface) cascaded with a tunnel initiator. For IPv4 data packets received from the local network, the CPE translates these packets, using the tunnel interface address as the source address, and then encapsulates the translated packet into IPv6, using the 4over6 concentrator IPv6 address as the destination address, the CPE's IPv6 address as source address. For IPv6 data packet received from the IPv6 network, the CPE performs decapsulation and IPv4 public-to-private translation. As to the CPE itself, it can use the public, tunnel interface address to communicate with the IPv4 Internet, and the private, IPv4 interface address to communicate with the local network.

5.3.3. non-NAT CPE as initiator

When the CPE doesn't perform a NAT function and end hosts in the local network get public IPv4 addresses allocated from the concentrator, the situation becomes a little complicated. To support dynamic address allocation in this situation, the CPE should act as an IPv4 DHCP relay, relaying the DHCP requests and replies between the host and the concentrator. Here the CPE's tunnel interface acts as the "upper" interface of the relay, i.e., the CPE uses an IPv4-in-IPv6 tunnel to forward DHCP messages to and receive DHCP messages from the concentrator. The static allocation method is similar to the former two case.

The remaining problem is what kind of IPv4 address does the CPE use. The address of tunnel interface is only used by the CPE itself, so it

could be a well-known IPv4 address, just like B4's configuration in DS-lite; Or the CPE could get a public IPv4 address from the concentrator and assigned it to the tunnel interface, in case that the CPE has its own IPv4 communication demand. As to the IPv4 interface connected to the local network, its address should be reachable in the local network, i.e., in the same range of the hosts' addresses. So the CPE should get a public IPv4 address from the concentrator and assigned it to the IPv4 interface, or the ISP could claim a specific IPv4 address from its 4over6 DHCPv4 pool and assign this unified address to every non-NAT CPE's IPv4 interface. In either case, the CPE should have its tunnel interface address and IPv4 interface address separated in different address ranges to avoid confusion. Here different strategies achieve different effects and consume IPv4 address to varying degrees. The authors would like to remain this topic as an open issue in this version of draft.

On data plan, for IPv4 data packets received from the local network, the CPE encapsulates them and forward them to IPv6 network. For IPv6 data packet received from the IPv6 network, the CPE performs decapsulation and forward them to IPv4 local network. No translation is requires since the end hosts use public addresses.

5.4. IPv4-IPv6 mapping maintaining methods

section 5.3 describes the address mapping maintaining with DHCP-driven updating, in which DHCP process on the concentrator triggers installation and deletion of IPv4-IPv6 address mappings. Another way to maintain the mapping is traffic snooping, i.e., record the address mapping when decapsulating IPv4-in-IPv6 data packets coming from 4over6 initiators. In this way, the mappings are installed based on the actual traffic rather than DHCP. However, the shortage of this method is that extra procedure is required to support inbound access. This happens when there's no mapping exists on the concentrator for an allocated IPv4 address, either because there's no outbound traffic from this IP yet or because the earlier-installed mapping has expired, while packets from the IPv4 Internet have already arrived on the concentrator and tried to reach the corresponding IP. To solve this problem, the 4over6 initiator need to send keepalive "pinhole" packets to the concentrator, or uses a protocol similar to PCP[I-D.ietf-pcp-base]. This draft recommends the DHCP-driven updating method since it's more accurate and controllable, and requires no extra procedure on the initiator.

If an operator chooses the DHCP-driven updating method, the concentrator need manual mapping configuration as well for static configured 4over6 initiators. The traffic snooping method works for both static and dynamic 4over6 initiators, though.

6. Technical advantages

Public 4over6 provides a method for users in IPv6 network to communicate with IPv4. In many scenarios, this can be viewed as an alternative to IPv6-IPv4 translation mechanisms which have well-known limitations described in [RFC4966] .

Since a 4over6 initiator uses a public IPv4 address, Public 4over6 supports full bidirectional communication between IPv4 Internet and hosts/IPv4 networks in IPv6 access network. In particular, it supports the servers in IPv6 network to provide IPv4 application service transparently.

Public 4over6 supports dynamic reuse of a single IPv4 address between multiple subscribers based on their dynamic requirement of communicating with IPv4 Internet. A subscriber will request a public IPv4 address for a period of time only when it need to communicate with IPv4 Internet. Besides, in the NATed CPE case, one public IPv4 address will be shared by the local network. So Public 4over6 can improve the reuse rate of IPv4 addresses.

Public 4over6 is suited for network users/ISPs which can still get/ provide public IPv4 addresses. Dual-stack lite is suited for network users/ISPs which can no longer get/provide public IPv4 addresses. By combining Public 4over6 and Dual-stack lite, the IPv4-over-IPv6 Hub & spoke problem can be well solved.

7. Acknowledgement

The authors would like to thank Alain Durand and Dan Wing for their valuable comments on this draft.

8. References

8.1. Normative References

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

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

[RFC4966] Aoun, C. and E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", RFC 4966, July 2007.

8.2. Informative References

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and F. Dupont, "Port Control Protocol (PCP)", draft-ietf-pcp-base-06 (work in progress), February 2011.

[I-D.ietf-softwire-ds-lite-tunnel-option]
Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual- Stack Lite", draft-ietf-softwire-ds-lite-tunnel-option-10 (work in progress), March 2011.

[I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-07 (work in progress), March 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

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

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

Chris Metz
Cisco Systems, Inc.
3700 Cisco Way
San Jose, CA 95134
USA

Email: chmetz@cisco.com

Olivier Vautrin
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, CA 94089
USA

Email: Olivier@juniper.net

Yiu L. Lee
Comcast

Email: yiu_lee@cable.comcast.com