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

Y. Cui
J. Wu
P. Wu
Tsinghua University
Q. Sun
C. Xie
China Telecom
C. Zhou
Huawei Technologies
July 11, 2011

Lightweight 4over6 in access network
draft-cui-softwire-b4-translated-ds-lite-01

Abstract

The dual-stack lite mechanism provide an IPv4 access method over IPv6 ISP network for end users. Dual-Stack Lite enables a broadband service provider to share IPv4 addresses among customers by combining IPv4-in-IPv6 tunnel and Carrier Grade NAT. However, in dual-stack lite. CGN has to maintain per-session address mapping, which could be a heavy burden, and produce high hardware cost as well as performance issue. This draft propose the lightweight 4over6 mechanism which moves the translation function from tunnel concentrator (AFTR) to initiators (B4s), and hence reduce the mapping scale on CGN to per-customer level. For translation usage, the mechanism will allocate port restrcted IPv4 addresses to initiators in a flexible way independent of IPv6 network in the middle.

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

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

Dual-stack lite technology provides IPv4 access method over IPv6 ISP network for broadband users. To deal with the incoming IPv4 exhaustion, dual-stack lite embeds the IPv4 address sharing functionality into the mechanism, by putting an IPv4 CGN on the tunnel concentrator (AFTR). The 44CGN solves the address shortage problem at the cost of maintaining per-session address mapping.

The common estimation of AFTR capacity is that one AFTR should serve thousands of end users. The huge amount of sessions from the users would cause a fairly high hardware cost, and could easily cause performance issues. Besides, if the CGN need to support source tracing, the volume of logging data will be really huge. Therefore it’s of great significance if we can reduce the amount of address mappings effectively.

The address mapping is used directly for IPv4 private-public translation. So in order to reduce the amount of address mappings on the concentrator, the straightforward approach is moving the translation from the concentrator to initiators. Then the concentrator will be only responsible for encapsulation and forwarding, and hence get rid of the burden of translation and state maintenance. Apparently an initiator can be capable of translating its own traffic since the volume is low. However, it does need the public address and port for translation allocated from the concentrator. [I-D.cui-softwire-host-4over6] has discussed the case where full IPv4 addresses are allocated over IPv6 for IPv4-over-IPv6 usage; this draft will cover the case where port space is divided and allocated, too.

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

Lightweight 4over6: lightweight 4over6 is an IPv4-over-IPv6 hub and spoke mechanism proposed in this document, which supports address sharing to deal with IPv4 address exhaustion, and places the IPv4 translation on the intiator side.

Lightweight 4over6 initiator (or "initiator" for short): the tunnel initiator in lightweight 4over6 mechanism. The lightweight 4over6 initiator could be a host directly connected to IPv6, or an dual-stack CPE in front of an IPv4 local network. It is responsible for IPv4 public-private translation besides tunnel encapsulation and decapsulation.

Lightweight 4over6 concentrator (or "concentrator" for short): the tunnel concentrator in lightweight 4over6 mechanism. The lightweight 4over6 concentrator connects the ISP IPv6 network and IPv4 Internet. It provides tunnel encapsulation and decapsulation but no IPv4 public-private translation.

4. Port Restricted IPv4 Address Allocation

As is described above, a lightweight 4over6 initiator needs the public address and port for stateful translation. It's obvious that individual address + port allocation when required is not efficient due to round-trip time delay and high signaling volume caused. Besides, that way can't save the mapping amount at all. A practical manner would be allocating a group of address + port at one time beforehand, i.e., allocating IPv4 address with a port range, which is known as "restricted address".

To our knowledge there're two methods which are probably suitable for restricted address allocation from concentrator to initiator. One is extending DHCP to support address allocation with port range embedded. [I-D.bajko-pripaddrassign] discusses this DHCP usage. In this special context, we need to build the DHCPv4 procedure over IPv6 [I-D.cui-softwire-dhcp-over-tunnel]. The other is extending PCP to support port range control. See [I-D.tsou-pcp-natcoord] for details. Adopting either method, An initiator can get port restricted IPv4 addresses allocated dynamically from the concentrator. Unlike stateless 4over6 solutions like [I-D.murakami-softwire-4rd], the port restricted address allocation in lightweight 4over6 has no requirement on IPv6 address format, i.e. IPv4 and IPv6 addressing remain separated. This is more close to today's address allocation mode and provides flexibility for ISPs to manage the access network.

5. Lightweight 4over6 Initiator Behavior

An lightweight 4over6 initiator should support either extended DHCP client, or extended PCP client, as is described above. When requiring IPv4 access, the initiator would run either client to get a dynamic port restricted IPv4 address, and renew/extend it when the lease/lifetime is about to expire.

The data plane functions of the initiator includes translation and encapsulation/decapsulation. An initiator runs a standard local NAT44 with the address pool consist of the allocated port restricted address. When sending out an IPv4 packet with private source address, it performs NAT44 function and translates the source address into public. Then it encapsulate the packet with concentrator's IPv6 address as destination IPv6 address, and forward it to the concentrator. When receiving an IPv4-in-IPv6 packet from the concentrator, the initiator decapsulates the IPv6 packet to get the IPv4 packet with public destination IPv4 address. Then it performs NAT44 function and translates the destination address into private.

For host initiator case, it's also feasible that the host doesn't run the local NAT and uses the allocated public IPv4 address directly. Then the host should guarantee that every port number in the packets sent out by itself falls into the allocated port range.

6. Lightweight 4over6 Concentrator Behavior

The lightweight 4over6 concentrator should support either an extended DHCPv4 server, or an extended PCP server, to allocate port restricted addresses. When accomplishing one such allocation, the concentrator simultaneously install a mapping entry into the mapping table. This entry consists of the public IPv4 address, the port range and initiator's IPv6 address. Its lifetime is set according to the allcation. It'll be used for encapsulation of inbound packets. This mapping entry would be deleted when the lifetime expires, and refresh its lifetime when the initiator renews/extends the allocation.

The data plane functions of the concentrator is purely encapsulation and decapsulation. When receiving an IPv4-in-IPv6 packet from an initiator, the concentrator decapsulates it and forwards it to IPv4 Internet. When receiving an IPv4 packet from the Internet, it uses the destination address and port from this packet to lookup the destination initiator's IPv6 address in the mapping table. Then the concentrator encapsulates this packet using the IPv6 address found in the table as IPv6 destination address, and forwards it to the correct initiator.

7. Mechanism Analysis

Compared with original dual-stack lite, lightweight 4over6 removes the translation burden from the concentrator and distribute the job to initiators on user-side. Also it decreases the state scale on concentrator from per-session level down to per-user level. This would significantly reduce the hardware cost of the concentrator, and the probability that the concentrator become the performance bottleneck. Leveraging lightweight 4over6, one concentrator can serve a lot more customers.

Besides, since this mechanism reduces the number of simultaneous address mappings of each customer on concentrator to one, it makes concentrator logging much more feasible. Moreover, locate the translation on user side eases the ALG and NAT referral problem since it'll be no different from the situation of local NAT in today's IPv4 network. Various solutions already exsit for quite a long time.

Lightweight 4over6 allocates port restricted address independent of the IPv6 network in the middle. No specific IPv6 address format is required. IPv4 and IPv6 addressing and routing remain separated. This is close to today's address allocation mode. The ISP can provision IPv4 in a flexible, on-demand way, as well as manage the native IPv6 network without the influence of IPv4-over-IPv6 requirements.

The costs of lightweight 4over6 for achieving all these benifits are lower IPv4 address utilization ratio and extra signaling behavior. The address multiplexing manner of port restricted address is relatively more static than the CGN manner. And dual-stack lite doesn't require address and port allocation between concentrator and initiators. When compared to stateless solutions, lightweight 4over6 still keeps per-user states rather than becoming purely stateless.

Besides, ICMP ping would become a challenge in port restricted address environment. The solution is to divide ICMP id field in the same way with dividing port space, i.e. each initiator will get the IMCP id range which is identical to the allocated port range. This way the initiator uses the allocated address and restricted id range when send out a ping. Hence ping "session" intiated from a lightweight 4over6 initiator can be processed correctly by the concentrator. The inbound ping would be left unsupported on the concentrator, which is the similar behavior of NAT/CGN. See details about ICMP ping porcessing in section 4.2 of [I-D.sun-v6ops-laft6].

8. References

[I-D.bajko-pripaddrassign]
Bajko, G., Savolainen, T., Boucadair, M., and P. Levis, "Port Restricted IP Address Assignment", draft-bajko-pripaddrassign-03 (work in progress), September 2010.

[I-D.cui-softwire-dhcp-over-tunnel]
Cui, Y., Wu, P., and J. Wu, "DHCPv4 Behavior over IP-IP tunnel", draft-cui-softwire-dhcp-over-tunnel-00 (work in progress), June 2011.

[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.

[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-11 (work in progress), May 2011.

[I-D.murakami-softwire-4rd]
Murakami, T. and O. Troan, "IPv4 Residual Deployment on IPv6 infrastructure - protocol specification", draft-murakami-softwire-4rd-00 (work in progress), July 2011.

[I-D.sun-v6ops-laft6]
Sun, Q. and C. Xie, "LAFT6: Lightweight address family transition for IPv6", draft-sun-v6ops-laft6-01 (work in progress), March 2011.

[I-D.tsou-pcp-natcoord]
Zhou, C., ZOU), T., Deng, X., Boucadair, M., and Q. Sun, "Using PCP To Coordinate Between the CGN and Home Gateway Via Port Allocation", draft-tsou-pcp-natcoord-03 (work in progress), July 2011.

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

Authors' Addresses

Yong Cui
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-62603059
Email: yong@csnet1.cs.tsinghua.edu.cn

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

Phone: +86-10-62785983
Email: jianping@cernet.edu.cn

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

Phone: +86-10-62785822
Email: weapon@csnet1.cs.tsinghua.edu.cn

Qiong Sun
China Telecom
Room 708, No.118, Xizhimennei Street
Beijing 100035
P.R.China

Phone: +86-10-58552936>
Email: sunqiong@ctbri.com.cn

Chongfeng Xie
China Telecom
Room 708, No.118, Xizhimennei Street
Beijing 100035
P.R.China

Phone: +86-10-58552116>
Email: xiechf@ctbri.com.cn

Cathy Zhou
Huawei Technologies
Section B, Huawei Industrial Base, Bantian Longgang
Shenzhen 518129
P.R.China

Phone: +86-10-58552116>
Email: cathyzhou@huawei.com

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

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

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

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.

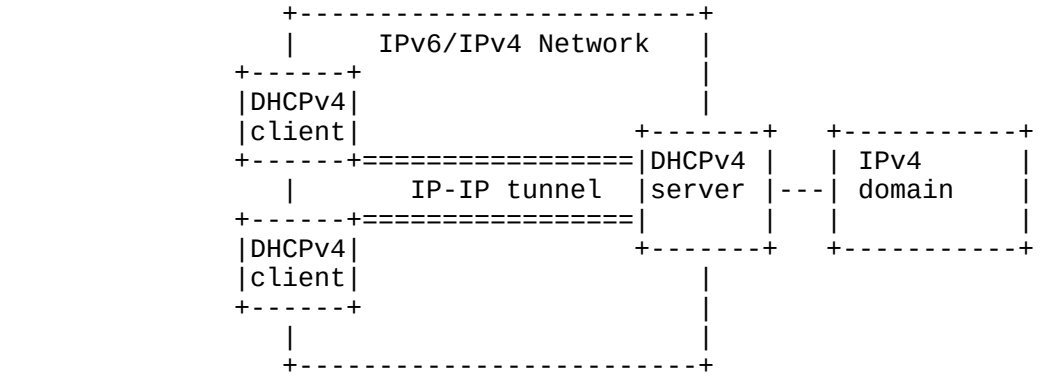


Figure 1 Scenario of DHCPv4 over tunnel

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.

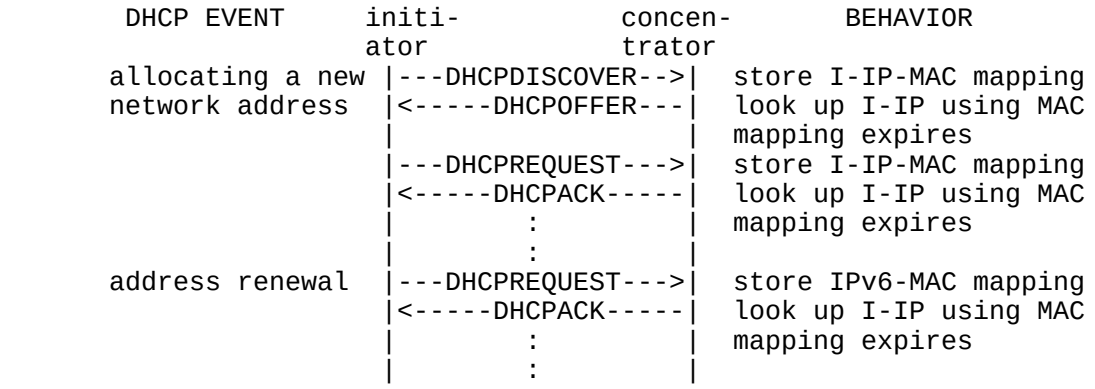


Figure 2 4over6 concentrator: DHCP behavior

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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2012

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

July 8, 2011

Public IPv4 over Access IPv6 Network
draft-cui-softwire-host-4over6-06

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 and 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 January 9, 2012.

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

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 networks 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 networks will communicate with network users sited in the IPv4-only part of Internet. This demand could eventually decrease with the general IPv6 adoption.

Network operators should provide IPv4 services to IPv6 users to satisfy their demand, 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 and requirements

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".

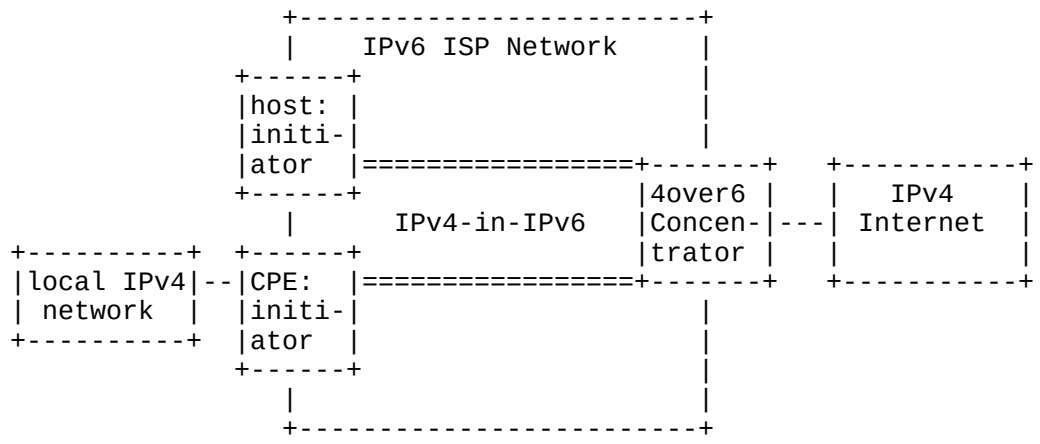


Figure 1 Public 4over6 scenario

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, the CPE can get a public IPv4 address and runs an IPv4 NAT for the local network. Here a local NAT is 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 need double translation along the path, one from IPv4 to IPv6 and the other from IPv6 back to IPv4. This would be quite complicated, especially in addressing. Contrarily a tunnel can achieve the IPv4-over-IPv6 traversing easily. That's the reason this draft follows the hub and spoke softwire model.

Moreover, the ISP probably would like to keep their IPv4 and IPv6 addressing and routing separated when provisioning IPv4 over IPv6. Then the ISP can manage the native IPv6 network more easily and independently, and also provision IPv4 in a flexible, on-demand way. The cost is that the concentrator needs to maintain per-user address mapping state, which would be described in detail.

4.2. Use cases

Public 4over6 can be applicable in several practical cases. The first one is that ISPs which still own enough IPv4 addresses switch to IPv6. The ISPs can deploy public 4over6 to preserve IPv4 service for the customers. This case is actually quite common. The majority of the wired end users today get Internet access with public IPv4 address. When their ISPs switch to IPv6, these users can still use the same amount of IPv4 addresses for IPv4 access. Public 4over6 can leveraging these addresses and offer tunneled IPv4 access.

The second case is ISPs which don't have enough IPv4 addresses any more switch to IPv6. For these ISPs, dual-stack lite is so far the most mature solution to provision IPv4 over IPv6. In dual-stack lite, end users use private IPv4 addresses, experience a 44CGN and hence some service degradation. As long as the end users use public IPv4 addresses, all CGN issues can be avoided and the IPv4 service can be full bi-directional. In other words, Public 4over6 can be deployed along with DS-lite, to provide a value-added service. Common users adopt DS-lite to communicate with IPv4 while high-end users adopt Public 4over6. The two mechanisms can actually be coupled easily.

There is also a special situation in the second case that the end users are IPv4 application servers. In this situation, public address brings significant convenience. The DNS registration can be

direct using dedicated address; the access of application clients can be straightforward with no translation; there's no need to reserve and maintain address mapping on the CGN, and no well-known port collision will come up. So it's better to have servers adopt Public 4over6 for IPv4 access when they're located in IPv6 network.

Following the principle of Public 4over6, it's also possible to achieve address multiplexing and save IPv4 addresses. There're already efforts on this subject, see [I-D.cui-softwire-b4-translated-ds-lite] and [I-D.sun-v6ops-laft6]. The basic idea is that instead of allocating a full IPv4 address to every end user, the ISP can allocate an IPv4 address with restricted port range to every end user.

Besides, the draft would like to be explicit about the scope of direct-connected host case and CPE case. The host case is clear: the host is directly connected to IPv6 network, but the protocol stack on the host support IPv4 too. As to the CPE case, this draft would like to only focus on the case that the local network behind the CPE is private IPv4. If the users want to run public IPv4 into the local network, then they can either run dual-stack in the local network and turn into host case(likely home LAN situation), or they can acquire address blocks from the ISP and build configured tunnel or softwire mesh[RFC5565] with the ISP network(likely enterprise network situation). TC can be implemented to be compatible with the latter case too, though.

5. Public 4over6 Mechanism

5.1. Address allocation and mapping maintenance

Public 4over6 can be generally considered as IPv4-over-IPv6 hub and 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 CPE case, every CPE will get one IPv4 address, which will be shared by hosts behind the CPE. The key problem here is IPv4 address allocation over IPv6 network, from ISP device(s) to separated 4over6 initiators.

There’re two possibilities here. One is DHCPv4 over IPv6, and the other is static configuration. DHCPv4 over IPv6 is achieved by performing DHCPv4 on IPv4-in-IPv6 tunnel between ISP device and 4over6 initiators. There do exist the DHCP encapsulation issue on server side, see details and solutions in [I-D.cui-softwire-dhcp-over-tunnel]. As to static configuration, 4over6 users and the ISP operators should negotiate beforehand to authorize the IPv4 address. Application servers usually falls into this case. Public 4over6 supports both address allocation manners. Actually, it is transparent to address allocation methods.

Along with IPv4 address allocation, Public 4over6 should maintain the IPv4-IPv6 address mappings on the concentrator. In this type of address mapping, the IPv4 address is the public IPv4 address allocated to a 4over6 initiator, and the IPv6 addresses is the initiator’s IPv6 address. This mapping is used to provide correct encapsulation destination address for the concentrator.

The initiator sends "pinhole" packets to the concentrator periodically, to install and renew the address mapping. A pinhole packet is an IPv4-in-IPv6 packet, which uses the concentrator’s IPv6 address as destination IPv6 address, the initiator’s IPv6 address as source IPv6 address, and the initiator’s IPv4 address as source IPv4 address. When the concentrator receives such a packet, it’ll resolve the IPv4 and IPv6 address information from the packet and trigger the mapping. Since any IPv4-in-IPv6 data packet from the initiator contains these exact informations, it can also serve as pinhole packet. Then dedicated pinhole packets are sent out when there’s no data packets. Another possible way to maintain the address mapping is to run PCP[I-D.ietf-pcp-base] while extending the protocol to support applying for a full address. The following sections describe the mechanism with the pinhole method.

5.2. 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].

5.2.1. Host initiator

When the initiator is a direct-connected host, it assigns the allocated public IPv4 address to its tunnel interface. The host uses this address for IPv4 communication. If this address is allocated through DHCP, the host should support DHCPv4 over tunnel. After the allocation, the host periodically sends pinhole packet to the concentrator to install the address mapping and keep it alive.

For IPv4 data traffic, the host performs the IPv4-in-IPv6 encapsulation and decapsulation on the tunnel interface. 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.

5.2.2. CPE initiator

The CPE case is quite similar to the host initiator case. The CPE assign the allocated IPv4 address to its tunnel interface. 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. After the allocation, the CPE periodically sends pinhole packet to the concentrator to install the address mapping and keep it alive.

On data plan, the 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 concentrator's 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 uses 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. 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 maintains the address mapping according to the initiators' demand. When receiving a pinhole packet from an initiator, the concentrator reads the IPv4 and IPv6 source addresses from the packet, install the mapping entry into the mapping table or renew it if it already exists. When the lifetime of a mapping entry expires, the concentrator deletes it from the table. So the initiator should send pinhole packet with an interval shorter than the lifetime of the mapping entry. The mapping entry is used to provide correct encapsulation destination address for concentrator encapsulation. As long as the entry exists in the table, the concentrator can encapsulate inbound IPv4 packets destined to the initiator, with the initiator's IPv6 address as IPv6 destination.

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, uses the IPv4 destination address in the packet to look up IPv6 destination address in the address mapping table. 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.

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 provides IPv4 access over IPv6 network while keeps IPv4-IPv6 addressing and routing separated. Therefore the ISP can manage the native IPv6 network independently without the influence of IPv4-over-IPv6 requirements, and also provision IPv4 in a flexible, on-demand way.

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 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 and 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.

[RFC5549] Le Faucheur, F. and E. Rosen, "Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop", RFC 5549, May 2009.

[RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, June 2009.

8.2. Informative References

[I-D.cui-softwire-b4-translated-ds-lite]
Cui, Y., Wu, J., and D. Wu, "B4 translated DS-lite enable AFTR to serve more B4s", draft-cui-softwire-b4-translated-ds-lite-00 (work in progress), October 2010.

[I-D.cui-softwire-dhcp-over-tunnel]
Cui, Y., Wu, P., and J. Wu, "DHCPv4 Behavior over IP-IP tunnel", draft-cui-softwire-dhcp-over-tunnel-00 (work in progress), June 2011.

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-13 (work in progress), July 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-11 (work in progress), May 2011.

[I-D.sun-v6ops-laft6]
Sun, Q. and C. Xie, "LAFT6: Lightweight address family transition for IPv6", draft-sun-v6ops-laft6-01 (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
One Comcast Center
Philadelphia, PA 19103
USA

Email: yiu_lee@cable.comcast.com

Softwire
Internet-Draft
Intended status: Standards Track
Expires: January 8, 2012

Y. Cui
J. Dong
P. Wu
Tsinghua University
July 7, 2011

Softwire Mesh Management Information Base(MIB)
draft-cui-softwire-mesh-mib-02

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular it defines objects for managing softwire mesh [RFC5565].

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 8, 2012.

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction

Softwire mesh framework RFC 5565 [RFC5565]is a tunneling mechanism which enables connectivity between islands of IPv4, IPv6 or dual-stack networks across single IPv4 or IPv6 backbone networks. In softwire mesh solution, extended multiprotocol-BGP (MP-BGP)is used to set up tunnels and advertise prefixes among address family border routers (AFBRs).

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular it defines objects for managing softwire mesh [RFC5565].

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

3. Terminology

This document uses terminology from softwire problem statement RFC 4925 [RFC4925] and softwire mesh framework RFC5565 [RFC5565].

4. Conventions

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 RFC 2119 [RFC2119].

5. Structure of the MIB Module

The softwire mesh MIB provides a way to configure and manage the softwire mesh objects through SNMP.

5.1. The swmSupportedTunnlTable Subtree

Since AFBR need to negotiate with BGP peer what kind of tunnel they use, it should firstly annouce what kind tunnels it supports. The swmSupportedTunnlTable subtree provides the infomation about what kind of tunnels the AFBR supports. According to section 4 of RFC 5512, current softwire mesh tunnel includes IP-IP, GRE or L2TPv3.

5.2. The swmEncapsTable Subtree

The swmEncapsTable subtree provides softwire mesh NLRI-NH information about the AFBR. It indicates which E-IP destination address will be encapsulated according to the arriving packet’s I-IP destination address.

5.3. The swmBGPNeighborTable Subtree

The subtree provides softwire mesh BGP neighbor information about the AFBR. It includes the address of softwire mesh BGP peer. It also includes what kind of tunnel that the AFBR would use to communicate with this BGP peer.

5.4. The swmMIBConformance Subtree

The subtree provides conformance information of MIB objects.

6. Relationship to Other MIB Modules

6.1. Relationship to the IF-MIB

The Interfaces MIB [RFC2863] defines generic managed objects for managing interfaces. Each logical interface (physical or virtual) has an ifEntry in the Interfaces MIB [RFC2863]. Tunnels are handled by creating a logical interface (ifEntry) for each tunnel. Softwire mesh tunnel also acts as a virtual interface, which has corresponding entries in IP Tunnel MIB and Interface MIB. Those corresponding entries are indexed by ifIndex.

The ifOperStatus in ifTable would be used to represents whether the mesh function of the AFBR has been started. During the BGP OPEN phase, if the softwire mesh capability is negotiated, the mesh function could be considered to be started, and ifOperStatus is "up", else the ifOperStatus is "down".

If it’s IPv4-over-IPv6 softwire mesh tunnel, the ifInUcastPkts represents the number of IPv6 packets which would be decapsulated to IPv4 in the virtual interface. The ifOutUcastPkts represents the number of IPv6 packets which have been encapsulated from IPv4

packets. Particularly, if these IPv4 packets need to be fragmented, the number counted here is the packets after fragmentation.

If it's IPv6-over-IPv4 softwire mesh tunnel, the ifInUcastPkts represents the number of IPv4 packets which would be decapsulated to IPv6 in the virtual interface. The ifOutUcastPkts represents the number of IPv4 packets which have been encapsulated from IPv6. Particularly, if these IPv6 packets need to be fragmented, the number counted here is the packets after fragmentation. Similar definition apply to other counting objects in ifTalbe.

6.2. Relationship to the IP Tunnel MIB

The IP Tunnel MIB [RFC4087] contains objects common to all IP tunnels, including softwire mesh. Additionally, tunnel encapsulation specific MIB (like what is defined in this document) extend the IP tunnel MIB to further describe encapsulation specific information.

In case of softwire mesh, Since it's a point to multi-point tunnel, we need to specify a ecapsulation talbe to support E-IP routing between AFBRs to achieve correct forwarding in E-IP networks and correct encapsulation on AFBRs. Each AFBR also needs to know information about remote BGP peers (AFBRs).

The implementation of the IP Tunnel MIB is required for softwire mesh. The tunnelIfEncapsMethod in the tunnelIfEntry should be set to softwireMesh("xx"), and corresponding entry in the softwire mesh MIB module will exist for every tunnelIfEntry with this tunnelIfEncapsMethod. The tunnelIfRemoteInetAddress must be set to 0.0.0.0 for IPv4 or :: for IPv6 because it's a point to multi-point tunnel.

Since tunnelIfAddressType in tunnelIfTable represents the type of address in the corresponding tunnelIfLocalInetAddress and tunnelIfRemoteInetAddress objects, we also can use the tunnelIfAddressType to specify the softwire mesh tunnel is IPv4-over-IPv6 or IPv6-over-IPv4 when the tunnelIfEncapsMethod is softwireMesh: When tunnelIfAddressType is IPv4, the encapsulation would be IPv6-over-IPv4; When tunnelIfAddressType is IPv6, the encapsulation would be IPv4-over-IPv6.

6.3. MIB modules required for IMPORTS

The following MIB module IMPORTS objects from SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], IF-MIB [RFC2863] and INET-ADDRESS-MIB [RFC4001].

7. Definitions

```
SOFTWIRE-MESH-MIB DEFINITIONS ::= BEGIN

IMPORTS
   TruthValue, TEXTUAL-CONVENTION
   TimeStamp
       FROM SNMPv2-TC

   OBJECT-GROUP, MODULE-COMPLIANCE
       FROM SNMPv2-CONF

   MODULE-IDENTITY, OBJECT-TYPE, mib-2, Unsigned32, Counter32,
   Counter64
       FROM SNMPv2-SMI

   IANAtunnelType          FROM IANAifType-MIB;

   InetAddress, InetAddressPrefixLength
       FROM INET-ADDRESS-MIB

swmMIB MODULE-IDENTITY
    LAST-UPDATED "201107070000Z"        -- July 7, 2011
    ORGANIZATION "Softwire Working Group"
    CONTACT-INFO "

         Yong Cui
         Email:  yong@csnet1.cs.tsinghua.edu.cn

         Jiang Dong
         Email:  dongjiang@csnet1.cs.tsinghua.edu.cn

         Peng Wu
         Email:  weapon@csnet1.cs.tsinghua.edu.cn

         Email comments directly to ther softwire WG Mailing
         List at softwires@ietf.org
     "

   DESCRIPTION
       "This MIB module contains managed object definitions for
        the softwire mesh framework."

   REVISION    "201107070000Z"
   DESCRIPTION
       "draft-02 version"
::= {transmission xxx} --xxx to be replaced with correct value

   -- swmSupportedTunnelTable
   swmSupportedTunnelTable OBJECT-TYPE
```

```
        SYNTAX      SEQUENCE OF swmSupportedTunnelEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "A table of objects that shows what kind of tunnels
            can be supported in the AFBR."
        ::= { swmMIB 1 }

    swmSupportedTunnelEntry  OBJECT-TYPE
        SYNTAX      swmSupportedTunnelEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "A set of objects that shows what kind of tunnels
            can be supported in the AFBR. Ef the AFBR supports
            several kinds of tunnel type, The
            swmSupportedTunnelTalbe would has sveral entries."
        INDEX { swmSupportedTunnelType }
        ::= { swmSupportedTunnelTable 1 }

    swmSupportedTunnelEntry ::=
        SEQUENCE {
            swmSupportedTunnelType              IANATunnelType
        }

    swmSupportedTunnelType OBJECT-TYPE
        SYNTAX      IANATunnelType
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Represents the kind of tunneling type that the AFBR
            support. "
        ::= { swmSupportedTunnelTypeEntry 1 }
    -- end of swmSupportedTunnelTable

    --swmEncapsTable
    swmEncapsTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF swmEncapsEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "A table of objects that display and control the
            softwire mesh encapsulation information."
        ::= { swmMIB 2 }

    swmEncapsEntry  OBJECT-TYPE
        SYNTAX      swmEncapsEntry
        MAX-ACCESS  not-accessible
```

```
        STATUS      current
        DESCRIPTION
            "A set of objects that display and control the
            softwire mesh encapsulation information."
        INDEX { ifIndex,
                swmEncapsIIPDst,
                swmEncapsIIPMask
              }
        ::= { swmEncapsTable 1 }

    swmEncapsEntry ::=
        SEQUENCE {
            swmEncapsIIPDst          InetAddress,
            swmEncapsIIPMask         InetAddressPrefixLength,
            swmEncapsEIPDst          InetAddress
        }

    swmEncapsIIPDst OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The destination I-IP address that decide which
            E-IP address will be encapsulated. The address Type
            is opposite to tunnelIfAddressType in tunnelIfTable."
        ::= { swmEncapsEntry 1 }

    swmEncapsIIPMask OBJECT-TYPE
        SYNTAX      InetAddressPrefixLength
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The prefix length of I-IP address."
        ::= { swmEncapsEntry 2 }

    swmEncapsEIPDst OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The E-IP address that will be encapsulated
             according to the I-IP address.The address Type
            is the same as tunnelIfAddressType in tunnelIfTable.
            Since the tunnelIfRemoteInetAddress in tunnelIfTable
            should be 0.0.0.0 or ::, swmEncapEIPDst is the
            destination address used in the outer IP header."
        ::= { swmEncapsEntry 3 }
    -- End of swmEncapsTable
```

```
-- swmBGPNeighborTable
swmBGPNeighborTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF swmBGPNeighborEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of objects that display the softwire mesh
        BGP neighbor information."
    ::= { swmMIB 3 }

swmBGPNeighborEntry  OBJECT-TYPE
    SYNTAX      swmBGPNeighborEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects that display the softwire mesh
        BGP neighbor information."
    INDEX {
            ifIndex,
            swmBGPNeighborInetAddress
                        }
    ::= { swmBGPNeighborTable 1 }

swmBGPNeighborEntry ::=
    SEQUENCE {
        swmBGPNeighborInetAddress        InetAddress,
        swmBGPNeighborTunnelType         IANATunnelType
    }

swmBGPNeighborInetAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The address of the ABFR's BGP neighbor. The
        address type is the same as tunnelIfAddressType
        in tunnelIfTable"
    ::= { swmBGPNeighborEntry 1 }

swmBGPNeighborTunnelType OBJECT-TYPE
    SYNTAX      IANATunnelType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Represents the kind of tunneling type that the
        AFBR used to comunication with the BGP neighbor"
    ::= { swmBGPNeighborEntry 2 }
-- End of swmBGPNeighborTable
```

```
     -- conformance information
      swmMIBConformance
                         OBJECT IDENTIFIER ::= { swmMIB 4 }
      swmMIBCompliances
                         OBJECT IDENTIFIER ::= { swmMIBConformance 1 }
      swmMIBGroups
                         OBJECT IDENTIFIER ::= { swmMIBConformance 2 }

     -- compliance statements
     swmMIBCompliance MODULE-COMPLIANCE
       STATUS current
       DESCRIPTION
           "Describes the requirements for conformance to the softwire
           mesh MIB."

       MODULE -- this module
       MANDATORY-GROUPS    {
                             swmSupportedTunnelGroup,
                             swmEncapsGroup,
                             swmBGPNeighborGroup }
       ::= { swmMIBCompliances 1 }

     swmSupportedTunnelGroup    OBJECT-GROUP
       OBJECTS {
           swmSupportedTunnelType
       }
       STATUS  current
       DESCRIPTION
           "The collection of objects which are used to show
           what kind of tunnel the AFBR supports."
       ::= { swmMIBGroups 1 }

     swmEncapsGroup    OBJECT-GROUP
       OBJECTS {
                   swmEncapsIIPDst,
                   swmEncapsIIPMask,
                   swmEncapsEIPDst
       }
       STATUS  current
       DESCRIPTION
           "The collection of objects which are used to display
           softwire mesh encapsulation information."
       ::= { swmMIBGroups 2 }

     swmBGPNeighborGroup    OBJECT-GROUP
       OBJECTS {
                   swmBGPNeighborInetAddress,
                   swmBGPNeighborTunnelType
```

```
    }
    STATUS  current
    DESCRIPTION
        "The collection of objects which are used to display
         softwire mesh BGP neighbor information."
    ::= { swmMIBGroups 3 }

        END
```

8. Security Considerations

The swmMIB module can be used for configuration of certain objects, and anything that can be configured can be incorrectly configured, with potentially disastrous results.

There are some management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability:

- Unauthorized changes to the swmVIf4over6Table and swmVIf6over4Table may disrupt the configuration of the 4over6 mtu and 6over4 mtu of the virtual interface.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

9. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry, and the following IANA-assigned tunnelType values recorded in the IANAtunnelType-MIB registry:

Descriptor	OBJECT IDENTIFIER value
swmMIB	{ transmission XXX }

```
IANAtunnelType ::= TEXTUAL-CONVENTION
    SYNTAX     INTEGER {

                softwireMesh ("XX")        -- softwire Mesh tunnel

                }
```

10. References

10.1. Normative References

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

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

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

[RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, April 2009.

[RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire

Mesh Framework", RFC 5565, June 2009.

10.2. Informative References

[RFC2223] Postel, J. and J. Reynolds, "Instructions to RFC Authors", RFC 2223, October 1997.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC4181] Heard, C., "Guidelines for Authors and Reviewers of MIB Documents", BCP 111, RFC 4181, September 2005.

10.3. URL References

[idguidelines] IETF Internet Drafts editor, "http://www.ietf.org/ietf/1id-guidelines.txt".

[idnits] IETF Internet Drafts editor, "http://www.ietf.org/ID-Checklist.html".

[xml2rfc] XML2RFC tools and documentation, "http://xml.resource.org".

[ops] the IETF OPS Area, "http://www.ops.ietf.org".

[ietf] IETF Tools Team, "http://tools.ietf.org".

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

Jiang Dong
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
EMail: dongjiang@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

V6OPS
Internet Draft
Intended status: Informational
Expires: January 9, 2012

X.Deng
T.Zheng
M.Boucadair
L.Wang
France Telecom
X.Huang
Q.Zhao
Y.Ma
BUPT

July 8, 2011

Implementing AplusP in the provider's IPv6-only network
draft-deng-v6ops-aplusp-experiment-results-01.txt

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 9, 2012.

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

This memo describes an implementation of A+P in a provider's IPv6-only network. It provides details of the implementation, network elements, configurations and test results as well. Besides traditional port range A+P, a scattered port sets flavour of A+P is also implemented and verified for the sake of distributing incoming ports among customers in a more discrete way. The test results consist of the application compatibility test, UPnP extension for A+P, port usage and BitTorrent behaviour with A+P.

This memo focuses on the IPv6 flavor of A+P.

Table of Contents

1. Introduction

A+P [draft-ymbk-aplusp-09] is a technique to share IPv4 addresses during the IPv6 transition period without requiring a NAT function in the provider's network. The main idea of A+P is treating some bits from the port number in the TCP/UDP header as additional end point

identifiers to extend the address field, thereby leaving a range of ports available to applications. This feature facilitates migration of networks to IPv6-only while offering the IPv4 connectivity services to customers, because the IPv4 address and the significant bits from the port range can be encoded in an IPv6 address and therefore transporting IPv4 traffic over IPv6 network by stateless IPv6 routing.

We have implemented A+P in a residential ADSL access network, where IPv6-only access network is provided over PPPoE. In this document, we describe the implementation environment including A+P IPv6 prefix format and network elements configurations, and results of application tests as well. The document focuses on the implementation of the SMAP function specified in [draft-ymbk-aplusp-09]:

- o Implement DHCPv6 options to retrieve an IPv4-embedded IPv6 address and a port range.
- o Support of those DHCPv6 options in both the DHCPv6 server side and the DHCPv6 client side.
- o Support of those DHCPv6 options in both the DHCPv6 server side and the DHCPv6 client side.

For extensive application tests results in A+P environment, please refer to [draft-boucadair-behave-bittorrent-portrange-02] and [draft-boucadair-port-range-01].

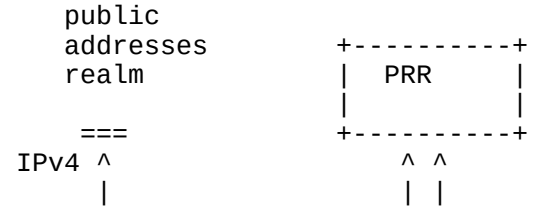
2. Terminology

This document makes use of the following terms:

- o PRR: Port Range Router
- o A+P CPE: A+P aware Customer Premise Equipment

3. Implementation environment

3.1. Environment Overview



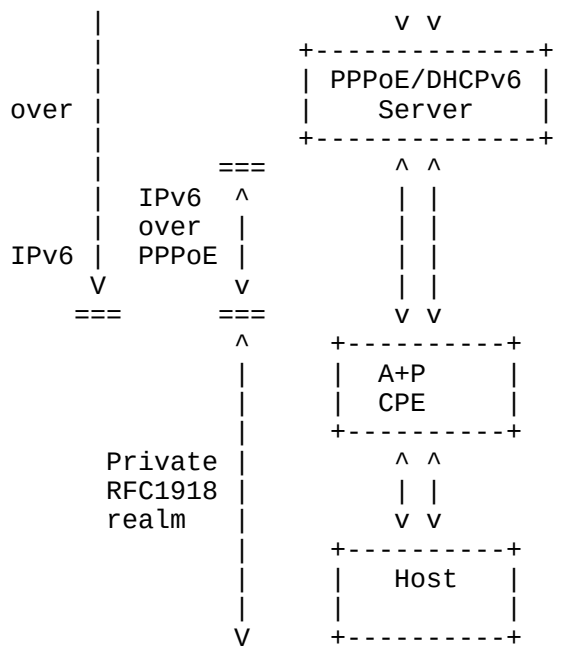


Figure 1 : Implementation Environment

We had developed both A+P home gate way function and Port Range Router (PRR) function on Linux platform and ported the home gate way function to a Linksys wrt 54G CPE, on which an openwrt 2.6.32 (based on Linux kernel) is running.

Figure 2 shows the Parameters of A+P CPE. IPv6 is provisioning over PPPoE to CPE while DHCPv6 server offers IPv6 prefix and A+P

parameters by extended options defined in [draft-boucadair-dhcpv6-shared-address-option].

Model	CPU Speed (MHz)	Flash (MB)	RAM (MB)	Wireless NIC	Wireless Standard	Wired Ports
Linksys WRT54GS	200	8	32	Broadcom (integrated)	11g	5

Figure 2 :Parameters of A+P CPE

3.2. Implementation and Configuration of A+P

Aplusp CPE, using Netfilter framework, the IPv4 port restricted NAT operation performed by CPE has been implemented by simply rules through iptables tool on Linux. After the port restriceted NAT operation, the IPv4 packets are sent to a TUN interface which is described as a virtual network interface in Linux. Using the IPv4-Embedded IPv6 address format defined in section 3.2.1, an IPv4-in-IPv6 encapsulation/decapsulation is performed by the TUN interface handler.

PRR, located in the interconnection point of the IPv6 network and IPv4 network, has been implemented with two main functions: 1) IPv4-in-IPv6 encapsulation/decapsulation; Like CPE, TUN driver is also used in PRR to achieve function IPv4-in-IPv6 encapsulation/decapsulation. 2) destination port based routing function, which is responsible for routing the IPv4 traffic originated from the IPv4 Internet to the Port Range restricted A+P CPE. Destination port based routing is implemented by generating IPv6 destination address, pre-assigned from IPv4 address and port range to each CPE, according to IPv4-Embedded IPv6 address format defined in section 3.2.1.

3.2.1. IPv4-Embedded IPv6 Address Format For A+P CPE

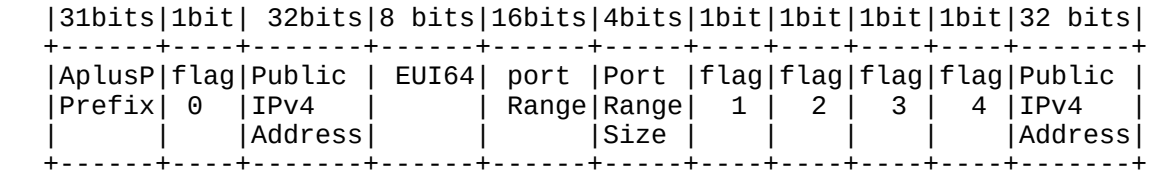


Figure 3 :IPv4-Embedded IPv6 address format

flag0: Is this address used by CPE or PRR?

flag1: Is address shared?

flag2: Is length of invariable present?

flag3: Is port range identifying sub network?

flag4: Reserved?

To facilitate test and experiment on AplusP solution, recently, we are considering release this AplusP implementation under open source license. For more implementation details, please refer to [Implementing A+P]

3.2.2. DHCPv6 Configurations

DHCPv6 options defined in [draft-boucadair-dhcpv6-shared-address-option] have been implemented. These options allow to configure a shared address together with a port range using DHCPv6.

3.2.3. Avoiding Fragmentation

Normally the TCP protocol stack will employ Maximum Segment Size (MSS) negotiation and/or Path Maximum Transmission Unit Discovery (PMTUD) to determine

the maximum packet size, and then try to send as large as possible datagram to achieve better throughput. However the IPv4-in-IPv6 encapsulation and the PPPoE header is very likly to cause a larger packet that exceeds the maximum MTU of the wire, and result in undesired fragmentation processing and decrease transmission

efficiency.

A simple solution is to enable iptables on A+P CPE to modify the MSS value of TCP session, using the command like "iptables -t mangle -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --set-mss DESIRED_MSS_VALUE". Here the DESIRED_MSS_VALUE is taken into account of common size of IPv4 header without options, common size of TCP header and size of basic IPv6 header and PPPoE header as well.

3.3. Implementing scattered Port Sets for A+P

3.3.1. Scattered Port Sets allocation mechanism

As described in [I-D.ietf-intarea-shared-addressing-issues], a bulk of incoming ports can be reserved as a centralized resource shared by all subscribers using a given restricted IPv4 address. In order to distribute incoming ports as scattered as possible among subscribers sharing the same restricted IPv4 address, other than allocating a continuous range of ports to per subscriber, a solution to distribute bulks of non-continuous ports among subscribers, which also takes port randomization of CPE NAT into account, because port randomization is one protection among others against blind attacks, is elaborated thereby.

On every restricted IPv4 address, according to port set size N, log2(N)bits are randomly chose as subscribers identification bits(s bit) among 1st and 16th bits. Take a sharing ration 1:32 for example, Figure 4 shows an example of 5bits (2nd, 5th, 7th, 9th, 11th) being chose as s bit.

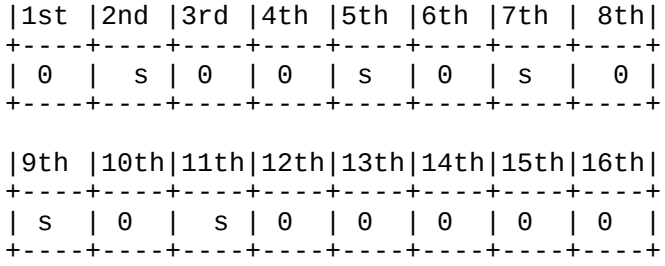


Figure 4 : An s bit selection example (on a sharing ration 1:32 address).

Subscriber ID pattern is then formed by setting all the s bits to 1

and other trivial bits to 0. Figure 5 illustrates an example of subscriber ID pattern which follows the s bit selection of figure 4. Note that the subscriber ID pattern can be different, ensured by the random s bit selection, per restricted IP address no matter whether the sharing ratio varies.

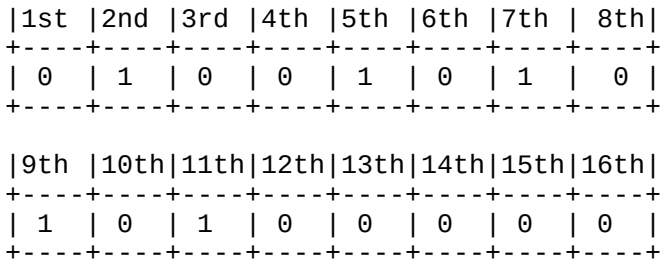


Figure 5 : A subscriber ID pattern example (on a sharing ration 1:32 address).

Subscribers ID value is then assigned by setting subscriber ID pattern bits (s bits shown in figure 4) to a unique customer value and setting other trivial bits to 1. An example of subscriber ID value, having a subscriber ID pattern shown in the figure 5 and a customer value 0, is shown in the figure 6.

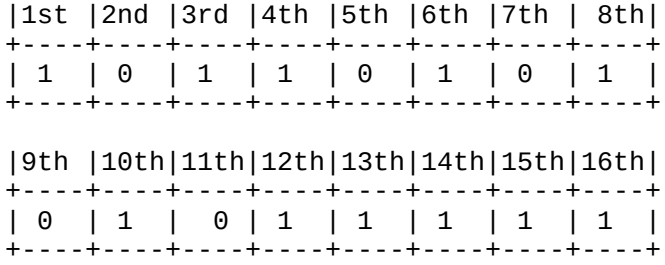


Figure 6 : A subscriber ID value example (customer value: 0)

Subscriber ID pattern and subscriber ID value together uniquely defines a restricted port set (Non-contiguous port sets or a contiguous port range, depends on Subscriber ID pattern and subscriber ID value) on a restricted IP address.

Pseudo-code shown in the figure 7 describes how to use subscriber ID pattern and subscriber ID value to implement a random ephemeral port selection function within the defined restricted port sets on a customer NAT.

```
      do{

          restricted_next_ephemeral = (random()|subscriber_ID_pattern)

                                      & subscriber_ID_value;

          if(five-tuple is unique)

          return restricted_next_ephemeral;

      }
```

Figure 7 : Random ephemeral port selection within the restricted port set

3.3.2. IPv4-Embedded IPv6 Address Format for Scattered Port Sets A+P CPE

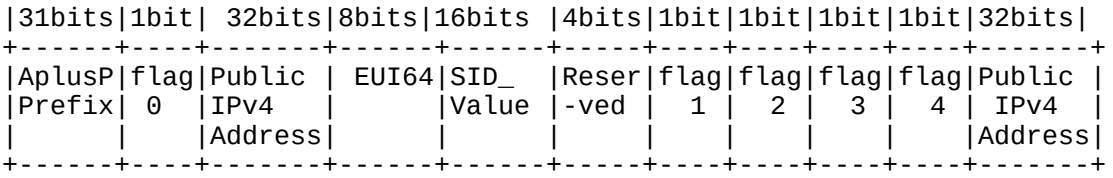


Figure 8 :IPv4-Embedded IPv6 address format

SID Value: Subscriber_ID_Value, which is unique for per subscriber sharing a given restricted IPv4 address. and has been allocated to each subscriber.

flag0: Is this address used by CPE or PRR?

flag1: Is address shared?

flag2: Is length of invariable present?

flag3: Is port range identifying sub network?

flag4: Reserved?

PRR maintains a mapping table, which consists of restricted IPv4 address and it's Subscriber ID Pattern. To form an IPv6 destination address for incoming packet, PRR could find the right SID Pattern according to a destination IPv4 address, and then apply a simple operation shown in the figure 9.

SID_Value = Destination_Port | (˜SID_Pattern).

Figure 9 :PRR calculates SID Value

3.3.3. Customize a scattered Ports Set A+P NAT on Linux

With a linux kernel 2.6.32.36, only one line of linux kernel code is changed, as shown in the figure5, and the same IPtables command line interface is used with the only one change of semantic that the original staring of port range becomes SID_Value and the ending port of a port range becomes SID_Pattern. The command line with iptables to configure a scattered Ports Set A+P is illustrated in the figure 11.

```
bool nf_nat_proto_unique_tuple(...)

  ...

//The Original code:

 //*portptr = htons(min + off % range_size);

 // was changed to:

  *portptr = htons((ntohs(off) | min ) & max );

   ...
```

Figure 10:Function of finding a unique 5-tuple for a scattered port sets A+P NAT

```
iptables -t nat -A POSTROUTING -o eth0 -p tcp -j SNAT --to-source
a.b.c.d: SID_Value-SID_Pattern --random

iptables -t nat -A POSTROUTING -o eth0 -p udp -j SNAT --to-source
a.b.c.d: SID_Value-SID_Pattern --random
```

Figure 11: IPtables commands for a scattered ports set A+P NAT

4. Application Tests and Experiments in A+P Environment

A set of well-known applications have been tested in this IPv6 flavor of A+P environment to access A+P impacts on them. The test results show that IPv6 flavor of A+P has the same impacts on applications as IPv4 flavor A+P does [draft-boucadair-port-range-01]. Web browsing (IE and Firefox), Email (Outlook), Instant message(MSN),Skype, Google Earth work normally with A+P. For more details, please refer to [draft-boucadair-port-range-01].

4.1. A+P Impacts on Applications

Application	A+P impacts
IE	None
Firefox	None
FTP(Passive mode)	None
FTP(Active mode)	require opening port forwarding
Skype	None
Outlook	None
Google Earth	None
BitComet	UPnP extensions may be required, when listening port is out of A+P range; other minor effects(see section 4.4)
uTorrent	UPnP extensions may be required, when listening port is out of A+P range; other minor effects(see section 4.4)

Live Messenger	None

Figure 12:Aplusp impacts on applications

For P2P (Peer-to-Peer) applications, when some of them listening on specific port to expect inbounding connection, it is likely to fail due to the listening port is out of A+P port range. Some UPnP extensions may be required to make P2P applications work properly with A+P. Other minor effects of A+P are discussed in section 4.4.

4.2. UPnP extension experiment

To make P2P application work properly with port restricted NAT , we have designed extensions including new variables, new errorcodes as well as new actions to UPnP 1.0, and have them implemented with [Emule], [open source UPnP SDK 1.0.4 for Linux] and [Linux UPnP IGD 0.92].

In figure 5, a new error code is proposed for the existing "AddPortMapping" action to explicitly indicate the situation that the requested external port is out of range.

ErrorCode	errorDescription	Description
728	ExternalPortOutOfRange	The external port is out of the port range assigned to this external interface

Figure 13:New ErrorCode for "AddPortMapping" action

New state variables have been introduced to reflect the valid port range. The definitions of these state variables are shown in figure 6.

Variable Name	Req. or Opt.	Data Type	Allowed Value	Default Value	Eng. Units
PortRangeLow	O	ui2	>=0	0	N/A
PortRangeHigh	O	ui2	<=65535	65535	N/A

Figure 14: New state variables for port range

Correspondingly, new actions, GetPortRangeLow and GetPortRangeHigh, defined to retrieve port range information are illustrated in figure 7. An IP address should be provided as argument to invoke the new actions, for the port range is associated with a specific IP address.

Action Name	Argument	Dir.	Related StateVariable
GetPortRangeLow	NewExternal IPAddress	IN	ExternalIPAddress
	NewPortRange Low	OUT	PortRangeLow
GetPortRangeHigh	NewExternal IPAddress	IN	ExternalIPAddress
	NewPortRange High	OUT	PortRangeHigh

Figure 15: New actions for port range

Please refer to [UPnP Extension] for more details of UPnP extension experiment in A+P.

4.3. Port Usage of Applications

Port consumptions of applications not only impact the deployment factor (i.e., port range size) for AplusP solution but also play an important role in determining the port limitation of per customer on

AFTR for Dual-Stack Lite.

Therefore we have also developed and deployed a Service Probe in our IPv6 network, which use IPv6 TCP socket to ask AplusP CPE for NAT session usage, and store AplusP NAT statistics in a Mysql database for further analysis of application behaviors in terms of port and session consumptions.

In figure 8, the maximum port usage of each application is the peak number of port consumption per second during the whole communication process. The duration time represents the total time from the first NAT binding entry being established to the last one being destroyed.

Application	Test case	Maximum port usage	Duration (seconds)
IE	browsing a news website	20-25	200
	browsing a video website	40-50	337
Firefox	browsing a news website	25-30	240
	browsing a video website	80-90	230
Chrome	browsing a news website	50-60	340
	browsing a video website	80-90	360
Android Chrome	browsing a news website	40-50	300
	browsing a video website	under 10	160
Google Earth	locating a place	30-35	240
Android Google Earth	locating a place	10-15	240
Skype	make a call	under 10	N/A
BitTorrent	downloading a file	200	N/A

Figure 16: Port usage of applications

4.4. BitTorrent Behaviour in A+P

[draft-boucadair-behave-bittorrent-portrange] provides an exhaustive testing report about the behaviour of BiTtorrent in an A+P architecture. [draft-boucadair-behave-bittorrent-portrange] describes the main behavior of BitTorrent service in an IP shared address environment. Particularly, the tests have been carried out on a testbed implementing [ID.boucadair-port-range] solution. The results are, however, valid for all IP shared address based solutions.

Two limitations were experienced. The first limitation occurs when two clients sharing the same IP address want to simultaneously retrieve the SAME file located in a SINGLE remote peer. This limitation is due to the default BitTorrent configuration on the remote peer which does not permit sending the same file to multiple ports of the same IP address. This limitation is mitigated by the fact that clients sharing the same IP address can exchange portions with each other, provided the clients can find each other through a common tracker, DHT, or Peer Exchange. Even if they can not, we observed that the remote peer would begin serving portions of the file automatically as soon as the other client (sharing the same IP address) finished downloading. This limitation is eliminated if the remote peer is configured with bt.allow_same_ip == TRUE.

The second limitation occurs when a client tries to download a file located on several seeders, when those seeders share the same IP address. This is because the clients are enforcing bt.allow_same_ip parameter to FALSE. The client will only be able to connect to one sender, among those having the same IP address, to download the file (note that the client can retrieve the file from other seeders having distinct IP addresses). This limitation is eliminated if the local client is configured with bt.allow_same_ip == TRUE, which is somewhat likely as those clients will directly experience better throughput by changing their own configuration.

Mutual file sharing between hosts having the same IP address has been checked. Indeed, machines having the same IP address can share files with no alteration compared to current IP architectures.

5. Security Considerations

TBD

6. IANA Considerations

This document includes no request to IANA.

7. Conclusion

Despite A+P introduces some impacts on existence applications, issues of P2P applications due to the port restricted NAT have been resolved by UPnP extension experiment in our test bed, and other issues are shared by other IP address sharing solutions. Therefore, from our work, it has been proved that deploying A+P in the Service Provider's IPv6 network during IPv6 transition period is feasible.

8. References

8.1. Normative References

[Implementing A+P]

Xiaoyu ZHAO.,"Implementing Public IPv4 Sharing in IPv6 Environment", ICCGI 2010

[UPnP Extension]

Xiaoyu ZHAO., "UPnP Extensions for Public IPv4 Sharing in IPv6 Environment", ICNS 2010

8.2. Informative References

[1] Faber, T., Touch, J. and W. Yue, "The TIME-WAIT state in TCP and Its Effect on Busy Servers", Proc. Infocom 1999 pp. 1573-1583.

[Fab1999] Faber, T., Touch, J. and W. Yue, "The TIME-WAIT state in TCP and Its Effect on Busy Servers", Proc. Infocom 1999 pp. 1573-1583.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119, March 1997.

[draft-ymbk-aplusp-09]

R. Bush., " The A+P Approach to the IPv4 Address Shortage", draft-ymbk-aplusp-09 (work in progress), February 17, 2011.

[draft-boucadair-dhcpv6-shared-address-option]

M. Boucadair., "Dynamic Host Configuration Protocol (DHCPv6) Options for Shared IP Addresses Solutions", draft-

boucadair-dhcpv6-shared-address-option-01 (work in progress), December 21, 2009

[draft-boucadair-port-range-01]

"IPv4 Connectivity Access in the Context of IPv4 Address Exhaustion", draft-boucadair-port-range-01(work in progress), January 30, 2009

[Emule]

http://www.emule-project.net/. [Accessed October 26, 2009]

[UPnP SDK 1.0.4 for Linux]

http://upnp.sourceforge.net/. [Accessed October 26, 2009].

[Linux UPnP IGD 0.92].

http://linuxigd.sourceforge.net/. [Accessed October 26, 2009].

[draft-boucadair-behave-bittorrent-portrange]

M. Boucadair.,"Behaviour of BitTorrent service in an IP Shared Address Environment", draft-boucadair-behave-bittorrent-portrange-02.txt

9. Acknowledgments

The experiments and tests described in this document have been explored, developed and implemented with help from Zhao Xiaoyu, Eric Burgey and JACQUENET Christian.

Thanks to Jan Zorz for comments.

Authors' Addresses

Xiaohong Deng
France Telecom
Hai dian district, 100190, Beijing,
China

Email: xiaohong.deng@orange-ftgroup.com

Mohamed BOUCADAIR
France Telecom
Rennes,35000 France

Email: mohamed.boucadair@orange-ftgroup.com

Lan Wang
France Telecom
Hai dian district, 100190, Beijing, China

Email: lan.wang@orange-ftgroup.com

Tao Zheng
France Telecom
Hai dian district, 100190, Beijing, China

Email: tao.zheng@orange-ftgroup.com

Xiaohong Huang
Beijing University of Post and Telecommunication
Email: huangxh@bupt.edu.cn

Qin Zhao
Beijing University of Post and Telecommunication
Email: zhaoqin.bupt@gmail.com

Yan MA
Beijing University of Post and Telecommunication
Email: mayan@bupt.edu.cn

softwire
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2012

R. Maglione
Telecom Italia
A. Durand
Juniper Networks

October 17, 2011

RADIUS Extensions for Dual-Stack Lite
draft-ietf-softwire-dslite-radius-ext-07

Abstract

Dual-Stack Lite is a solution to offer both IPv4 and IPv6 connectivity to customers which are addressed only with an IPv6 prefix. Dual-Stack Lite requires to pre-configure the Dual-Stack Lite Address Family Transition Router (AFTR) tunnel information on the Basic Bridging BroadBand (B4) element. In many networks, the customer profile information may be stored in Authentication Authorization and Accounting (AAA) servers while client configurations are mainly provided through Dynamic Host Configuration Protocol (DHCP). This document specifies a new Remote Authentication Dial In User Service (RADIUS) attribute to carry Dual-Stack Lite Address Family Transition Router Tunnel name; the RADIUS attribute is defined based on the equivalent DHCPv6 OPTION_AFTR_NAME option. This RADIUS attribute is meant to be used between the RADIUS Server and the Network Access Server (NAS), it is not intended to be used directly between the Basic Bridging BroadBand element and the RADIUS Server.

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 April 19, 2012.

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction

Dual-Stack Lite [RFC6333] is a solution to offer both IPv4 and IPv6 connectivity to customers which are addressed only with an IPv6 prefix (no IPv4 address is assigned to the attachment device). One of its key components is an IPv4-over-IPv6 tunnel, but a Dual-Stack-Lite Basic Bridging BroadBand (B4) will not know if the network it is attached to offers Dual-Stack Lite support, and if it did, would not know the remote end of the tunnel to establish a connection.

To inform the Basic Bridging BroadBand (B4) of the Address Family Transition Router's (AFTR) location, a Fully Qualified Domain Name (FQDN) may be used. Once this information is conveyed, the presence of the configuration indicating the AFTR's location also informs a host to initiate Dual-Stack Lite (DS-Lite) service and become a Softwire Initiator.

[RFC6334] specifies a DHCPv6 option which is meant to be used by a Dual-Stack Lite client (Basic Bridging BroadBand element, B4) to discover its Address Family Transition Router (AFTR) name. In order to be able to populate such option the DHCPv6 Server must be pre-provisioned with the Address Family Transition Router (AFTR) name.

In Broadband environments, customer profile may be managed by AAA servers, together with user Authentication, Authorization, and Accounting (AAA). Remote Authentication Dial In User Service (RADIUS) protocol [RFC2865] is usually used by AAA Servers to communicate with network elements. [I-D.ietf-radext-ipv6-access] describes a typical broadband network scenario in which the Network Access Server (NAS) acts as the access gateway for the users (hosts or CPEs) and the NAS embeds a DHCPv6 Server function that allows it to locally handle any DHCPv6 requests issued by the clients.

Since the DS-Lite AFTR information can be stored in AAA servers and the client configuration is mainly provided through Dynamic Host Configuration Protocol (DHCP) running between the NAS and the requesting clients, a new RADIUS attribute is needed to send AFTR information from AAA server to the NAS.

This document aims at defining a new RADIUS attribute to be used for carrying the DS-Lite Tunnel Name, based on the equivalent DHCPv6 option already specified in [RFC6334]

2. Terminology

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].

The terms DS-Lite Basic Bridging BroadBand element (B4) and the DS-Lite Address Family Transition Router element (AFTR) are defined in [RFC6333]

3. DS-Lite Configuration with RADIUS and DHCPv6

The Figure 1 illustrates how the RADIUS protocol and DHCPv6 work together to accomplish DS-Lite configuration on the B4 element when a PPP Session is used to provide connectivity to the user.

The Network Access Server (NAS) operates as a client of RADIUS and as DHCP Server for DHC protocol. The NAS initially sends a RADIUS Access Request message to the RADIUS server, requesting authentication. Once the RADIUS server receives the request, it validates the sending client and if the request is approved, the AAA server replies with an Access Accept message including a list of attribute-value pairs that describe the parameters to be used for this session. This list MAY also contain the AFTR Tunnel Name. When the NAS receives a DHCPv6 client request containing the DS-Lite tunnel Option, the NAS SHALL use the name returned in the RADIUS DS-Lite-Tunnel-Name attribute to populate the DHCPv6 OPTION_AFTR_NAME option in the DHCPv6 reply message.

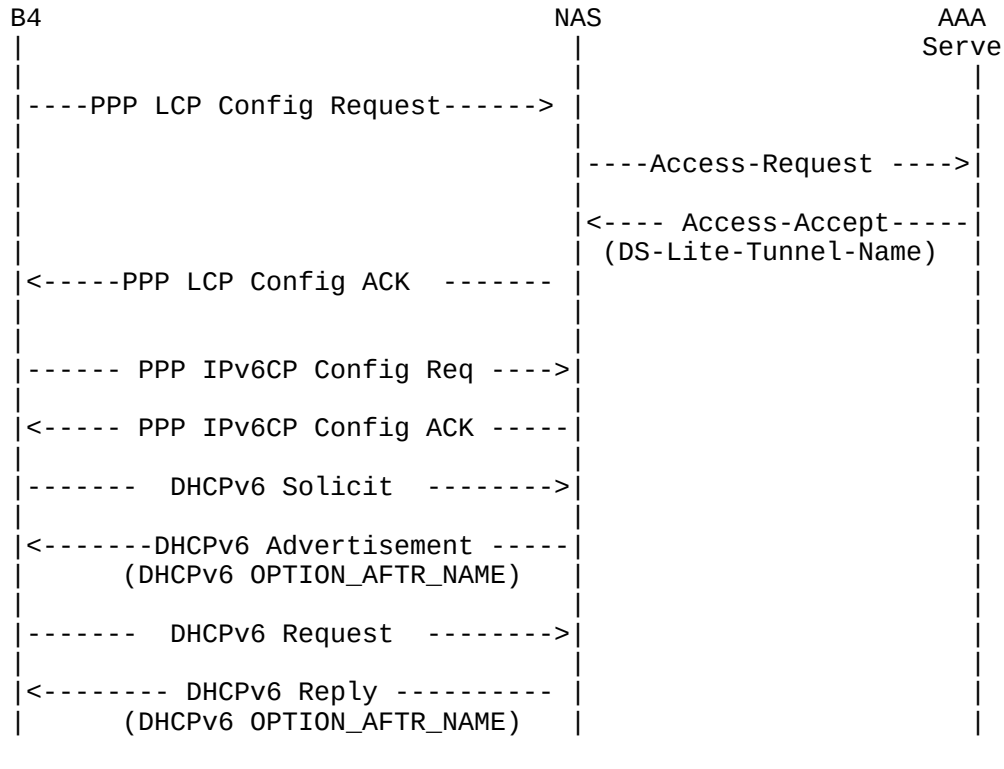


Figure 1: RADIUS and DHCPv6 Message Flow for a PPP Session

The Figure 2 illustrates how the RADIUS protocol and DHCPv6 work together to accomplish DS-Lite configuration on the B4 element when an IP Session is used to provide connectivity to the user.

The only difference between this message flow and previous one is that in this scenario the interaction between NAS and AAA/ RADIUS Server is triggered by the DHCPv6 Solicit message received by the NAS from the B4 acting as DHCPv6 client, while in case of a PPP Session the trigger is the PPP LCP Config Request message received by the NAS.

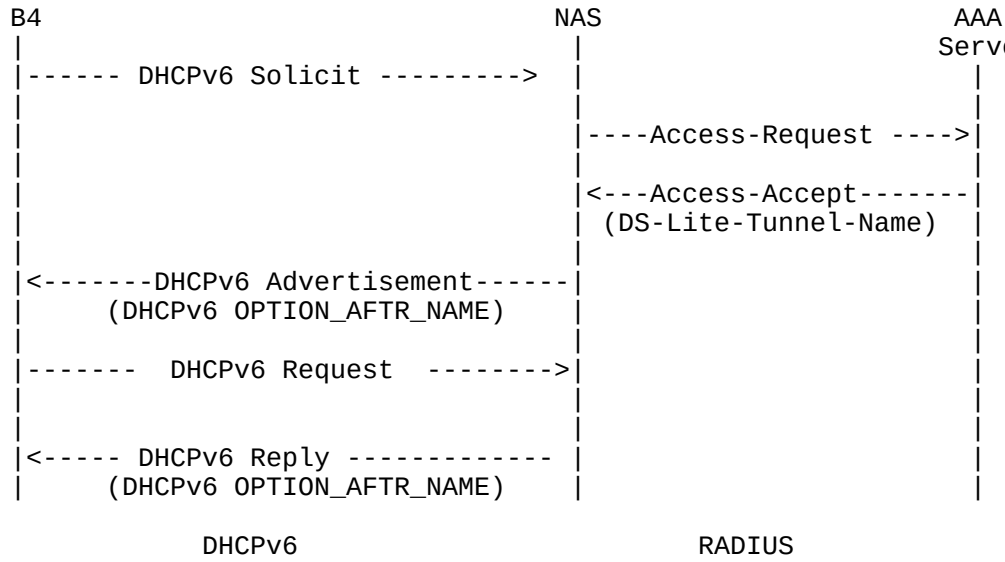


Figure 2: RADIUS and DHCPv6 Message Flow for an IP Session

In the scenario depicted in Figure 2 the Access-Request packet contains a Service-Type attribute with the value Authorize Only (17), thus according to [RFC5080] the Access-Request packet MUST contain a State attribute.

After receiving the DS-Lite-Tunnel-Name in the initial Access-Accept the NAS MUST store the received AFTR Tunnel Name locally. When the B4 sends a DHCPv6 Renew message to request an extension of the lifetimes for the assigned address or prefix, the NAS does not have to initiate a new Access-Request towards the AAA server to request the AFTR tunnel name. The NAS retrieves the previously stored AFTR tunnel name and uses it in its reply.

According to [RFC3315] if the DHCPv6 server to which the DHCPv6 Renew message was sent at time T1 has not responded, the DHCPv6 client initiates a Rebind/Reply message exchange with any available server. In this scenario the NAS receiving the DHCPv6 rebind message MUST initiate a new Access-Request towards the AAA server. The NAS MAY include the DS-Lite-Tunnel-Name attribute in its Access-Request.

If the NAS does not receive the DS-Lite-Tunnel-Name attribute in the Access-Accept it MAY fallback to a pre-configured default tunnel name, if any. If the NAS does not have any pre-configured default tunnel name or if the NAS receives an Access-Reject, the IPv4 over IPv6 tunnel cannot be established, thus the B4 element has only IPv6 connectivity.

4. RADIUS Attribute

This section specifies the format of the new RADIUS attribute.

4.1. DS-Lite-Tunnel-Name

Description

The DS-Lite-Tunnel-Name RADIUS attribute contains a Fully Qualified Domain Name that refers to the AFTR the client is requested to establish a connection with. The NAS SHALL use the name returned in the RADIUS DS-Lite-Tunnel-Name attribute to populate the DHCPv6 OPTION_AFTR_NAME option [RFC6334]

This attribute MAY be used in Access-Request packets as a hint to the RADIUS server; for example if the NAS is pre-configured with a default tunnel name, this name MAY be inserted in the attribute. The RADIUS server MAY ignore the hint sent by the NAS and it MAY assign a different AFTR tunnel name.

If the NAS includes the DS-Lite-Tunnel-Name attribute, but the AAA server does not recognize it, this attribute MUST be ignored by the AAA Server.

If the NAS does not receive DS-Lite-Tunnel-Name attribute in the Access-Accept it MAY fallback to a pre-configured default tunnel name, if any. If the NAS does not have any pre-configured default tunnel name, the tunnel can not be established.

If the NAS is pre-provisioned with a default AFTR tunnel name and the AFTR tunnel name received in Access-Accept is different from the configured default, then the AFTR tunnel name received in the Access-Accept message MUST be used for the session.

If the NAS cannot support the received AFTR tunnel name for any reason, the tunnel SHOULD NOT be established.

When the Access-Request is triggered by a DHCPv6 Rebind message if the AFTR tunnel name received in the Access-Accept is different from the currently used one for that session, the NAS MUST force the B4 to re-establish the tunnel using the new AFTR name received in the Access-Accept message.

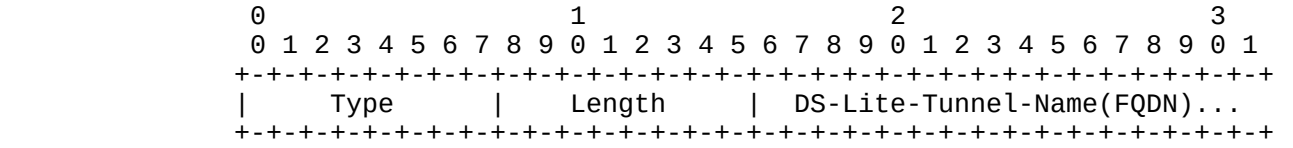
If an implementation includes the Change-of-Authorization (CoA) messages [RFC5176], they could be used to modify the current established DS-Lite tunnel. When the NAS receives a CoA Request message containing the DS-Lite-Tunnel-Name attribute, the NAS MUST send a Reconfigure message to a B4 to inform the B4 that the NAS has

new or updated configuration parameters and that the B4 is to initiate a Renew/Reply or Information-request/Reply transaction with the NAS in order to receive the updated information.

Upon receiving an AFTR tunnel name different from the currently used one, the B4 MUST terminate the current DS-Lite tunnel and the B4 MUST establish a new DS-LITE tunnel with the specified AFTR.

The DS-Lite-Tunnel-Name RADIUS attribute MAY be present in Accounting-Request records where the Acct-Status-Type is set to Start, Stop or Interim-Update. The DS-Lite-Tunnel-Name RADIUS attribute MUST NOT appear more than once in a message.

A summary of the DS-Lite-Tunnel-Name RADIUS attribute format is shown below. The fields are transmitted from left to right.



Type:

TBA1 for DS-Lite-Tunnel-Name.

Length:

This field indicates the total length in octets of this attribute including the Type, the Length fields and the length in octets of the DS-Lite-Tunnel-Name field

DS-Lite-Tunnel-Name:

A single Fully Qualified Domain Name of the remote tunnel endpoint, located at the DS-Lite AFTR.

As the DS-Lite-Tunnel-Name attribute is used to populate the DHCPv6 OPTION_AFTR_NAME option, the DS-Lite-Tunnel-Name field is formatted as required in DHCPv6 (Section 8 of [RFC3315] "Representation and Use of Domain Names"). Briefly, the format described is using a single octet noting the length of one DNS label (limited to at most 63 octets), followed by the label contents. This repeats until all labels in the FQDN are exhausted, including a terminating zero-length label. Any updates to Section 8 of [RFC3315] also apply to encoding of this field.

The data type of DS-Lite-Tunnel-Name RADIUS attribute is a string with opaque encapsulation, according to section 5 of [RFC2865]

5. Table of attributes

The following tables provide a guide to which attributes may be found in which kinds of packets, and in what quantity.

Access-Request	Access-Accept	Access-Reject	Challenge	Accounting-Request	#	Attribute
0-1	0-1	0	0	0-1	TBA1	DS-Lite-Tunnel-Name

CoA-Request	CoA-ACK	CoA-NACK	#	Attribute
0-1	0	0	TBA1	DS-Lite-Tunnel-Name

The following table defines the meaning of the above table entries.

0	This attribute MUST NOT be present in packet.
0+	Zero or more instances of this attribute MAY be present in packet.
0-1	Zero or one instance of this attribute MAY be present in packet.

6. Security Considerations

This document has no additional security considerations beyond those already identified in [RFC2865] for RADIUS protocol and in [RFC5176] for CoA messages.

[RFC6333] discusses Dual-Stack Lite related security issues.

7. IANA Considerations

This document requests the allocation of a new Radius attribute types from the IANA registry "Radius Attribute Types" located at http://www.iana.org/assignments/radius-types

DS-Lite-Tunnel-Name - TBA1

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.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[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.

[RFC5080] Nelson, D. and A. DeKok, "Common Remote Authentication Dial In User Service (RADIUS) Implementation Issues and Suggested Fixes", RFC 5080, December 2007.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

[RFC6334] Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite", RFC 6334, August 2011.

8.2. Informative References

[I-D.ietf-radext-ipv6-access] Lourdelet, B., Dec, W., Sarikaya, B., Zorn, G., and D. Miles, "RADIUS attributes for IPv6 Access Networks", draft-ietf-radext-ipv6-access-05 (work in progress), July 2011.

[RFC5176] Chiba, M., Dommety, G., Eklund, M., Mitton, D., and B. Aboba, "Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)", RFC 5176, January 2008.

Authors' Addresses

Roberta Maglione
Telecom Italia
Via Reiss Romoli 274
Torino 10148
Italy

Phone:
Email: roberta.maglione@telecomitalia.it

Alain Durand
Juniper Networks
1194 North Mathilda Avenue
Sunnyvale, CA 94089-1206
USA

Phone:
Fax:
Email: adurand@juniper.net
URI:

SOFTWIRE WG
Internet-Draft
Intended status: Standards Track
Expires: October 30, 2012

F. Brockners
S. Gundavelli
Cisco
S. Speicher
Deutsche Telekom AG
D. Ward
Cisco

April 28, 2012

Gateway Initiated Dual-Stack Lite Deployment
draft-ietf-softwire-gateway-init-ds-lite-08

Abstract

Gateway-Initiated Dual-Stack lite (GI-DS-lite) is a variant of Dual-Stack lite (DS-lite) applicable to certain tunnel-based access architectures. GI-DS-lite extends existing access tunnels beyond the access gateway to an IPv4-IPv4 NAT using softwires with an embedded context identifier that uniquely identifies the end-system the tunneled packets belong to. The access gateway determines which portion of the traffic requires NAT using local policies and sends/receives this portion to/from this softwire.

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 October 30, 2012.

Copyright Notice

Copyright (c) 2012 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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Overview

Gateway-Initiated Dual-Stack lite (GI-DS-lite) is a variant of the Dual-Stack lite (DS-lite) [RFC6333], applicable to network architectures which use point to point tunnels between the access device and the access gateway. The access gateway in these models is designed to serve large numbers of access devices. Mobile architectures based on Mobile IPv6 [RFC6275], Proxy Mobile IPv6 [RFC5213], or GTP [TS29060], as well as broadband architectures based on PPP or point-to-point VLANs as defined by the Broadband Forum [TR59] and [TR101] are examples for this type of architecture.

The DS-lite approach leverages IPv4-in-IPv6 tunnels (or other tunneling modes) for carrying the IPv4 traffic from the customer network to the Address Family Transition Router (AFTR). An established softwire between the AFTR and the access device is used for traffic forwarding purposes. This turns the inner IPv4 address irrelevant for traffic routing and allows sharing private IPv4 addresses [RFC1918] between customer sites within the service provider network.

Similar to DS-lite, GI-DS-lite enables the service provider to share public IPv4 addresses among different customers by combining tunneling and NAT. It allows multiple access devices behind the access gateway to share the same private IPv4 address [RFC1918]. Rather than initiating the tunnel right on the access device, GI-DS-lite logically extends the already existing access tunnels beyond the access gateway towards the Address Family Transition Router (AFTR) using a tunneling mechanism with semantics for carrying context state related to the encapsulated traffic. This approach results in supporting overlapping IPv4 addresses in the access network, requiring no changes to either the access device, or to the access architecture. Additional tunneling overhead in the access network is also omitted. If e.g., GRE based encapsulation mechanisms is chosen, it allows the network between the access gateway and the AFTR to be either IPv4 or IPv6 and provides the operator to migrate to IPv6 in incremental steps.

2. Conventions

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].

The following abbreviations are used within this document:

AFTR: Address Family Transition Router. An AFTR combines IP-in-IP tunnel termination and IPv4-IPv4 NAT.

AD: Access Device. It is the end host, also known as the mobile node in mobile architectures.

CID: Context Identifier

DS-lite: Dual-stack lite

GI-DS-lite: Gateway-initiated DS-lite

NAT: Network Address Translator

SW: Softwire [RFC4925]

SWID: Softwire Identifier

3. Gateway Initiated DS-Lite

The section provides an overview of Gateway Initiated DS-Lite (GI-DS-lite). Figure 1 outlines the generic deployment scenario for GI-DS-lite. This generic scenario can be mapped to multiple different access architectures, some of which are described in Appendix A.

In Figure 1, access devices (AD-1 and AD-2) are connected to the Gateway using some form of tunnel technology and the same is used for carrying IPv4 (and optionally IPv6) traffic of the access device. These access devices may also be connected to the Gateway over point-to-point links. The details on how the network delivers the IPv4 address configuration to the access devices are specific to the access architecture and are outside the scope of this document. With GI-DS-lite, Gateway and AFTR are connected by a softwire [RFC4925]. The softwire is identified by a softwire identifier (SWID). The SWID does not need to be globally unique, i.e. different SWIDs could be used to identify a softwire at the different ends of a softwire. The form of the SWID depends on the tunneling technology used for the softwire. The SWID could e.g. be the endpoints of a GRE-tunnel or a VPN-ID, Section 6 for details. A Context-Identifier (CID) is used to multiplex flows associated with the individual access devices onto the softwire. Deployment dependent, the flows from a particular AD can be identified using either the source IP-address or an access tunnel identifier. Local policies at the Gateway determine which part of the traffic received from an access device is tunneled over the softwire to the AFTR. The combination of CID and SWID must be unique between gateway and AFTR to identify the flows associated with an AD. The CID is typically a 32-bit wide identifier and is assigned

by the Gateway. It is retrieved either from a local or remote (e.g. AAA) repository. Like the SWID, the embodiment of the CID depends on the tunnel mode used and the type of the network connecting Gateway and AFTR. If, for example GRE [RFC2784] with "GRE Key and Sequence Number Extensions" [RFC2890] is used as softwire technology, the network connecting Gateway and AFTR could be either IPv4-only, IPv6-only, or a dual-stack IP network. The CID would be carried within the GRE-key field. Section 6 for details on different softwire types supported with GI-DS-lite.

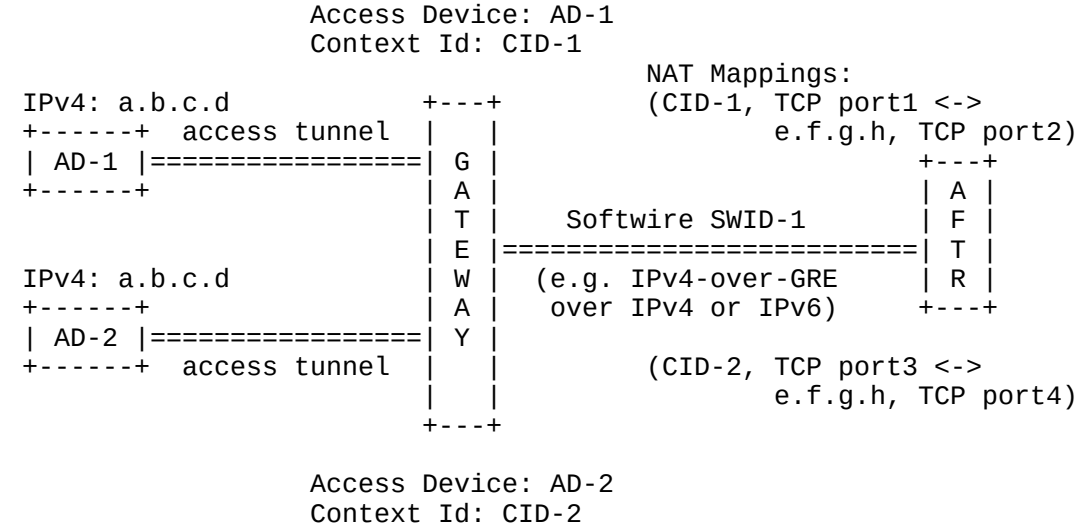


Figure 1: Gateway-initiated dual-stack lite reference architecture

The AFTR combines softwire termination and IPv4-IPv4 NAT. The NAT binding of the AD's address could be assigned autonomously by the AFTR from a local address pool, configured on a per-binding basis (either by a remote control entity through a NAT control protocol or through manual configuration), or derived from the CID (e.g., the CID, in case 32-bit wide, could be mapped 1:1 to an external IPv4-address). A simple example of a translation table at the AFTR is shown in Figure 2. The choice of the appropriate translation scheme for a traffic flow can take parameters such as destination IP-address, incoming interface, etc. into account. The IP-address of the AFTR, which, depending on the transport network between the Gateway and the AFTR, will either be an IPv6 or an IPv4 address, is configured on the Gateway. A variety of methods, such as out-of-band mechanisms, or manual configuration apply.

Softwire-Id/Context-Id/IPv4/Port	Public IPv4/Port
SWID-1/CID-1/a.b.c.d/TCP-port1	e.f.g.h/TCP-port2
SWID-1/CID-2/a.b.c.d/TCP-port3	e.f.g.h/TCP-port4

Figure 2: Example translation table on the AFTR

GI-DS-lite does not require a 1:1 relationship between Gateway and AFTR, but more generally applies to (M:N) scenarios, where M Gateways are connected to N AFTRs. Multiple Gateways could be served by a single AFTR. AFTRs could be dedicated to specific groups of access-devices, groups of Gateways, or geographic regions. An AFTR could, but does not have to be co-located with a Gateway.

4. Protocol and related Considerations

- Depending on the embodiment of the CID (e.g. for GRE-encapsulation with GRE-key), the NAT binding entry maintained at the AFTR, which reflects an active flow between an access device inside the network and a node in the Internet, SHOULD be extended to include the CID and the identifier of the softwire (SWID).

- When creating an IPv4 to IPv4 NAT binding for an IPv4 packet flow received from the Gateway over the softwire, the AFTR SHOULD associate the CID with that NAT binding. It SHOULD use the combination of CID and SWID as the unique identifier and use those parameters in the NAT binding entry.

- When forwarding a packet to the access device, the AFTR SHOULD obtain the CID from the NAT binding associated with that flow. E.g., in case of GRE-encapsulation, it SHOULD add the CID to the GRE Key and Sequence number extension of the GRE header and tunnel it to the Gateway.

- On receiving any packet from the softwire, the AFTR SHOULD obtain the CID from the incoming packet and use it for performing the NAT binding look up and for performing the packet translation before forwarding the packet.

- The Gateway, on receiving any IPv4 packet from the access device SHOULD lookup the CID for that access device. In case of GRE encapsulation it can for example add the CID to the GRE Key and Sequence number extension of the GRE header and tunnel it to the

AFTR.

- On receiving any packet from the softwire, the Gateway SHOULD obtain the CID from the packet and use it for making the forwarding decision.

- When encapsulating an IPv4 packet, Gateway and AFTR SHOULD use its Diffserv Codepoint (DSCP) to derive the DSCP (or MPLS Traffic-Class Field in case of MPLS) of the softwire.

5. Softwire Management and related Considerations

The following are the considerations related to the operational management of the softwire between AFTR and Gateway.

- The softwire between the Gateway and the AFTR MAY be created at system startup time, OR dynamically established on-demand. Deployment dependent, Gateway and AFTR can employ OAM mechanisms such as ICMP, BFD [RFC5880], or LSP ping [RFC4379] for softwire health management and corresponding protection strategies.

- The softwire peers MAY be provisioned to perform policy enforcement, such as for determining the protocol-type or overall portion of traffic that gets tunneled, or for any other quality of service related settings. The specific details on how this is achieved or the types of policies that can be applied are outside the scope for this document.

- The softwire peers SHOULD use the correct path MTU value for the tunnel path between the access gateway and the AFTR. This value MAY be statically configured at softwire creation time, or dynamically discovered using the standard path MTU discovery techniques.

- A Gateway and an AFTR can have multiple softwires established between them (e.g. to separate address domains, provide for load-sharing etc.).

6. Softwire Embodiments

Deployment and requirements dependent, different tunnel technologies apply for the softwire connecting Gateway and AFTR. GRE encapsulation with GRE-key extensions, MPLS VPNs [RFC4364], or plain IP-in-IP encapsulation can be used. Softwire identification and Context-ID depend on the tunneling technology employed:

- GRE with GRE-key: SWID is the tunnel identifier of the GRE tunnel between the GW and the AFTR. The CID is the GRE-key associated with the AD.

- MPLS VPN: The SWID is a generic identifier which uniquely identifies the VPN at either the Gateway or AFTR. Depending on whether the Gateway or AFTR are acting as customer edge (CE) or, provider edge (PE), the SWID could e.g. be an attachment circuit identifier, an identifier representing the set of VPN route labels pointing to the routes within the VPN, etc. The AD's IPv4-address is the CID. For a given VPN, the AD's IPv4 address must be unique.

- IPv4/IPv6-in-MPLS: The SWID is the top MPLS label. CID might be the next MPLS label in the stack, if present, or the IP address of the AD.

- IPv4-in-IPv4: SWID is the outer IPv4 source address. The AD's IPv4 address is the CID. For a given outer IPv4 source address, the AD's IPv4 address must be unique.

- IPv4-in-IPv6: SWID is the outer IPv6 source address. If the AD's IPv4 address is used as CID, the AD's IPv4 address must be unique. If the IPv6-Flow-Label [RFC6437] is used as CID, the IPv4 addresses of the ADs may overlap. Given that the IPv6-Flow-Label is 20-bit wide, which is shorter than the recommended 32-bit CID, large scale deployments may require additional scaling considerations. In addition, one should ensure sufficient randomization of the IP-Flow-Label to avoid possible interference with other uses of the IP-Flow-Label, such as Equal Cost Multipath (ECMP) support.

Figure 3 gives an overview of the different tunnel modes as they apply to different deployment scenarios. "x" indicates that a certain deployment scenario is supported. The following abbreviations are used:

- IPv4 address

  * "up": Deployments with "unique private IPv4 addresses" assigned to the access devices are supported.

  * "op": Deployments with "overlapping private IPv4 addresses" assigned to the access devices are supported.

  * "s": Deployments where all access devices are assigned the same IPv4 address are supported.

o Network-type

* "v4": Gateway and AFTR are connected by an IPv4-only network

* "v6": Gateway and AFTR are connected by an IPv6-only network

* "v4v6": Gateway and AFTR are connected by a dual stack network, supporting IPv4 and IPv4.

* "MPLS": Gateway and AFTR are connected by a MPLS network

Softwire	IPv4 address			Network-type			
	up	op	s	v4	v6	v4v6	MPLS
GRE with GRE-key	x	x	x	x	x	x	
MPLS VPN	x	x					x
IPv4/IPv6-in-MPLS	x	x	x				x
IPv4-in-IPv4	x			x			
IPv4-in-IPv6	x				x		
IPv4-in-IPv6 w/ FL	x	x	x		x		

Figure 3: Tunnel modes and their applicability

7. IANA Considerations

This specification does not require any IANA actions.

8. Security Considerations

The approach specified in this document allows the use of Dual-stack lite for tunnel-based access architectures. Rather than initiating the tunnel from the access device, GI-DS-lite logically extends the already existing access tunnel beyond the access gateway towards the Address Family Transition Router, and builds a virtual softwire between the AFTR and the access device. This approach requires the use of an additional context identifier in the AFTR and at the access gateway, which is required for making IP packet forwarding decisions.

A packet when received with an Incorrect context identifier at the access gateway/AFTR will result in associating the packet to an incorrect access device. Therefore, care must be taken to ensure an IP packet tunneled between the access gateway and the AFTR is carried

with the context identifier of the access device associated with that IP packet. The context identifier is not carried from the access device and it is not possible for one access device to claim the context identifier of some other access device. However, It is possible an on-path attacker between the access gateway and the AFTR can potentially modify the context identifier in the packet, resulting in association of the packet to an incorrect access device. This threat is no different from an on-path attacker modifying the source/destination address of an IP packet. However, this threat can be prevented by enabling IPsec security with integrity protection turned on, between the access gateway and the AFTR, that will ensure the correct binding of the context identifier and the inner packet. This specification does not require any other new security considerations other than those specified in dual-stack lite specification [RFC6333], and in the security considerations specified for the given access architecture, such as Proxy Mobile IPv6, leveraging this transitioning scheme.

9. Acknowledgements

The authors would like to acknowledge the discussions on this topic with Mark Grayson, Jay Iyer, Kent Leung, Vojislav Vucetic, Flemming Andreasen, Dan Wing, Jouni Korhonen, Teemu Savolainen, Parviz Yegani, Farooq Bari, Mohamed Boucadair, Vinod Pandey, Jari Arkko, Eric Voit, Yiu L. Lee, Tina Tsou, Guo-Liang Yang, Cathy Zhou, Olaf Bonness, Paco Cortes, Jim Guichard, Stephen Farrell, Pete Resnik, Ralph Droms.

10. References

10.1. Normative References

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

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

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, September 2000.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private

Networks (VPNs)", RFC 4364, February 2006.

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.

[RFC5213] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.

[RFC5555] Soliman, H., "Mobile IPv6 Support for Dual Stack Hosts and Routers", RFC 5555, June 2009.

[RFC5844] Wakikawa, R. and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6", RFC 5844, May 2010.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

[RFC6275] Perkins, C., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, November 2011.

10.2. Informative References

[I-D.draft-ietf-dime-nat-control] Brockners, F., Bhandari, S., Singh, V., and V. Fajardo, "Diameter NAT Control Application", August 2009.

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

[TR101] Broadband Forum, "TR-101: Migration to Ethernet-Based DSL Aggregation", April 2006.

[TR59] Broadband Forum, "TR-059: DSL Evolution - Architecture Requirements for the Support of QoS-Enabled IP Services", September 2003.

[TS23060] "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS); Service description; Stage 2.", 2009.

[TS23401] "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access.", 2009.

[TS29060] "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP), V9.1.0", 2009.

Appendix A. GI-DS-lite deployment

A.1. Connectivity establishment: Example call flow

Figure 4 shows an example call flow - linking access tunnel establishment on the Gateway with the softwire to the AFTR. This simple example assumes that traffic from the AD uses a single access tunnel and that the Gateway will use local polices to decide which portion of the traffic received over this access tunnel needs to be forwarded to the AFTR.

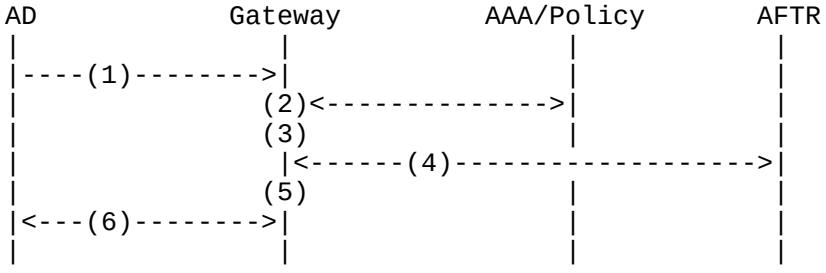


Figure 4: Example call flow for session establishment

1. Gateway receives a request to create an access tunnel endpoint.

2. The Gateway authenticates and authorizes the access tunnel. Based on local policy or through interaction with the AAA/Policy system the Gateway recognizes that IPv4 service should be provided using GI-DS-lite.

3. The Gateway creates an access tunnel endpoint. The access tunnel links AD and Gateway.

4. (Optional): The Gateway and the AFTR establish a control session between each other. This session can for example be used to

exchange accounting or NAT-configuration information. Accounting information could be supplied to the Gateway, AAA/Policy, or other network entities which require information about the externally visible address/port pairs of a particular access device. The Diameter NAT Control Application [I-D.draft-ietf-dime-nat-control] could for example be used for this purpose.

5. The Gateway allocates a unique CID and associates those flows received from the access tunnel that need to be tunneled towards the AFTR with the softwire linking Gateway and AFTR. Local forwarding policy on the Gateway determines which traffic will need to be tunneled towards the AFTR.

6. Gateway and AD complete the access tunnel establishment (depending on the procedures and mechanisms of the corresponding access network architecture this step can include the assignment of an IPv4 address to the AD).

A.2. GI-DS-lite applicability: Examples

The section outlines deployment examples of the generic GI-DS-lite architecture described in Section 3.

- Mobile IP based access architectures: In a DSMIPv6 [RFC5555] based network scenario, the Mobile IPv6 home agent will implement the GI-DS-lite Gateway function along with the dual-stack Mobile IPv6 functionality.

- Proxy Mobile IPv6 based access architectures: In a PMIPv6 [RFC5213] scenario the local mobility anchor (LMA) will implement the GI-DS-lite Gateway function along with the PMIPv6 IPv4 support [RFC5844] functionality.

- GTP based access architectures: 3GPP TS 23.401 [TS23401] and 3GPP TS 23.060 [TS23060] define mobile access architectures using GTP. For GI-DS-lite, the PDN-Gateway/GGSN will also assume the Gateway function.

- Fixed WiMAX architecture: If GI-DS-lite is applied to fixed WiMAX, the ASN-Gateway will implement the GI-DS-lite Gateway function.

- Mobile WiMAX: If GI-DS-lite is applied to mobile WiMAX, the home agent will implement the Gateway function.

- PPP-based broadband access architectures: If GI-DS-lite is applied to PPP-based access architectures the Broadband Remote Access Server (BRAS) or Broadband Network Gateway (BNG) will implement

the GI-DS-lite Gateway function.

- In broadband access architectures using per-subscriber VLANs the BNG will implement the GI-DS-lite Gateway function.

Authors' Addresses

Frank Brockners
Cisco
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Sri Gundavelli
Cisco
170 West Tasman Drive
SAN JOSE, CA 95134
USA

Email: sgundave@cisco.com

Sebastian Speicher
Deutsche Telekom AG
Landgrabenweg 151
BONN, NORDRHEIN-WESTFALEN 53277
Germany

Email: sebastian.speicher@telekom.de

David Ward
Cisco
170 West Tasman Drive
SAN JOSE, CA 95134
USA

Email: wardd@cisco.com

Softwire
Internet-Draft
Intended status: Informational
Expires: January 12, 2012

Y. Lee
Comcast
R. Maglione
Telecom Italia
C. Williams
MCSR Labs
C. Jacquenet
M. Boucadair
France Telecom
July 11, 2011

Deployment Considerations for Dual-Stack Lite
draft-lee-softwire-dslite-deployment-02

Abstract

This document discusses the deployment issues and describes requirements for the deployment and operation of Dual-Stack Lite. This document describes the various deployment considerations and applicability of the Dual-Stack Lite architecture.

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

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Overview

Dual-stack Lite (DS-Lite) [I-D.ietf-softwire-dual-stack-lite] is a transition technique that enable operators to multiplex public IPv4 addresses while provisioning only IPv6 to users. DS-Lite is designed to address the IPv4 depletion issue and allow the operators to upgrade their network incrementally to IPv6. DS-Lite combines IPv4-in-IPv6 tunnel and NAT44 to share a public IPv4 address more than one user. This document discusses various deployment considerations for DS-Lite by operators.

2. AFTR Deployment Considerations

2.1. Interface Consideration

Address Family Transition Router (AFTR) is the function deployed inside the operator's network. AFTR can be a standalone device or embedded into a router. AFTR is the IPv4-in-IPv6 tunnel termination point and the NAT44 device. It is deployed at the IPv4-IPv6 network border where the tunnel interface is IPv6 and the NAT interface is IPv4. Although an operator can configure a dual-stack interface for both functions, we recommended to configure two individual interfaces (i.e. one dedicated for IPv4 and one dedicated for IPv6) to segregate the functions.

2.2. MTU Considerations

DS-Lite is part tunneling protocol. Tunneling introduces some additional complexity and has a risk of MTU. With tunneling comes additional header overhead that implies that the tunnel's MTU is smaller than the raw interface MTU. The issue that the end user may experience is that they cannot download Internet pages or transfer files using File Transfer Protocol (FTP).

To mitigate the tunnel overhead, the access network could increase the MTU size to account the necessary tunnel overhead which is the size of an IPv6 header. If the access network MTU size is fixed and cannot be changed, the B4 element and the AFTR must support fragmentation.

2.3. Fragmentation

The IPv4-in-IPv6 tunnel is between B4 and AFTR. When a host behind the B4 element communicates to a server, both the host and the server are not aware of the tunnel. They may continue to use the maximum MTU size for communication. In fact, the IPv4 packet isn't over-sized, it is the v6 encapsulation that may cause the oversize. So

the tunnel points are responsible to handle the fragmentation. In general, the Tunnel-Entry Point and Tunnel-Exist Point should fragment and reassemble the oversized datagram. If the DF is set, the B4 element should send an ICMP "Destination Unreachable" with "Fragmentation Needed and Don't Fragment was Set" and drop the packet. If the DF is not set, the B4 element should fragment the IPv6 packet after the encapsulation. This mechanism is transport protocol agnostic and works for both UDP and TCP.

[editor note: Should we drop the IPv4 packet when DF is set?]

2.4. Lawful Intercept Considerations

Because of its IPv4-in-IPv6 tunneling scheme, interception of IPv4 sessions in DS-Lite architecture must be performed on the AFTR. Subjects can be uniquely identified by the IPv6 address assigned to the B4 element. Operator must associate the B4's IPv6 address and the public IPv4 address and port used by the subject.

Monitoring of a single subject may mean statically mapping the subject to a certain range of ports on a single IPv4 address, to remove the need to follow dynamic port mappings. A single IPv4 address, or some range of ports for each address, might be set aside for monitoring purposes to simplify such procedures. This requires to create a static mapping of a B4 element's IPv6 address to an IPv4 address that used for lawful intercept.

2.5. Logging at the AFTR

The timestamped logging is essential for tracing back specific users when a problem is identified from the outside of the AFTR. Such a problem is usually a misbehaving user in the case of a spammer or a DoS source, or someone violating a usage policy. Without time-specific logs of the address and port mappings, a misbehaving user stays well hidden behind the AFTR.

In DS-Lite framework, each B4 element is given a unique IPv6 address. The AFTR uses this IPv6 address to identify the B4 element. Thus, the AFTR must log the B4's IPv6 address and the IPv4 information. There are two types of logging: (1) Source-Specific Log and (2) Destination-Specific Log. For Source-Specific Log, the AFTR must timestamped log the B4's IPv6 address, transport protocol, source IPv4 address after NAT-ed, and source port. If a range of ports is dynamically assigned to a B4 element, the AFTR may create one log per range of ports to aggregate number of log entries. For Destination-Specific Log, the AFTR must timestamped log the B4's IPv6 address, transport protocol, source IPv4 address after NAT-ed, source port, destination address and destination port. The AFTR must log every

session from the B4 elements. No log aggregation can be performed. When using Destination-Specific Log, the operator must be careful of the large number of log entries created by the AFTR.

2.6. Blacklisting a shared IPv4 Address

AFTR is a NAT device. It shares a single IPv4 address with multiple users. [I-D.ietf-intarea-shared-addressing-issues] discusses many considerations when sharing address. When a pubic IPv4 address is blacklisted, this may affect multiple users and there is no effective way for the B4 element to notify the AFTR an IP address is blacklisted. It is recommended the server must no longer rely solely on IP address to identify an abused user. The server should combine the information stored in the transport layer (e.g. source port) and application layer (e.g. HTTP) to identify an abused user. [I-D.boucadair-intarea-nat-reveal-analysis] analyzes different approaches to identify a user in a shared address environment.

2.7. AFTR's Policies

There are two types of AFTR polices: (1) Outgoing Policies and (2) Incoming Policies. The outgoing policies must be implemented on the AFTR's internal interface connected to the B4 elements. The policies may include ACL and QoS settings. For example: the AFTR may only accept B4's connections originated from the IPv6 prefixes provisioned in the AFTR. The AFTR may also give priority to the packets marked by certain DSCP values. The AFTR may also limit the rate of port creation from a single B4's IPv6 address. Outgoing policies could be applied to individual B4 element or a set of B4 elements.

The incoming policies must be implemented on the AFTR's external interface connected to the IPv4 network. Similar to the outgoing policies, the policies may include ACL and QoS settings. Incoming policies are usually more general and globally applied to all users rather than individual user.

2.8. AFTR Impacts on Accounting Process in Broadband Access

DS-Lite introduces challenges to IPv4 accounting process. In a typical DSL/Broadband access scenario where the Residential Gateway (RG) is acting as a B4 element, the BNAS is the IPv6 edge router which connects to the AFTR. The BNAS is normally responsible for IPv6 accounting and all the subscriber manager functions such as authentication, authorization and accounting. However, given the fact that IPv4 traffic is encapsulated into an IPv6 packet at the B4 level and only decapsulated at the ATFR level, the BNAS can't do the IPv4 accounting without examining the inner packet. AFTR is the next logical place to perform IPv4 accounting, but it will potentially

introduce some additional complexity because the AFTR does not have detailed customer identity information.

The accounting process at the AFTR level is only necessary if the Service Provider requires separate per user accounting records for IPv4 and IPv6 traffic. If the per user IPv6 accounting records, collected by the BNAS, are sufficient, the additional complexity to be able to implement IPv4 accounting at the ATFR level is not required. It is important to consider that, since the IPv4 traffic is encapsulated in IPv6 packets, the data collected by the BNAS for IPv6 traffic already contain the total amount of traffic (i.e. IPv6 plus IPv4).

Even if detailed accounting records collection for IPv4 traffic may not be required, in some scenarios it would be useful for a Service Provider, to have inside the RADIUS Accounting packet, generated by the BNAS for the IPv6 traffic, a piece of information that can be used to identify the AFTR that is handling the IPv4 traffic for that user. This can be achieved by adding into the IPv6 accounting records the RADIUS attribute information specified in [I-D.ietf-softwire-dslite-radius-ext]

2.9. Reliability Considerations of AFTR

The service provider can use techniques to achieve high availability such as various types of clusters to ensure availability of the IPv4 service. High availability techniques include the cold standby mode. In this mode the AFTR states are not replicated from the Primary AFTR to the Backup AFTR. When the Primary AFTR fails, all the existing established sessions will be flushed out. The internal hosts are required to re-establish sessions to the external hosts. Another high availability option is the hot standby mode. In this mode the AFTR keeps established sessions while failover happens. AFTR states are replicated from the Primary AFTR to the Backup AFTR. When the Primary AFTR fails, the Backup AFTR will take over all the existing established sessions. In this mode the internal hosts are not required to re-establish sessions to the external hosts. The final option is to deploy a mode in between these two whereby only selected sessions such as critical protocols are replicated. Criteria for sessions to be replicated on the backup would be explicitly configured on the AFTR devices of a redundancy group.

2.10. Strategic Placement of AFTR

The public IPv4 addresses are pulled away from the customer edge to the outside of the centralized AFTR where many customer networks can share a single public IPv4 address. The AFTR architecture design is mostly figuring out the strategic placement of each AFTR to best use

the capacity of each public IPv4 address without oversubscribing the address or overtaxing the AFTR itself.

AFTR is a tunnel concentrator, B4 traffic must pass through the AFTR to reach the IPv4 Internet. Managing tunnels and NAT could be resource intensive, so the placement of the AFTR would affect the traffic flows in the access network and have operation implications. In general, there are two placements to deploy AFTR. Model One is to deploy the AFTR in the edge of network to cover a small region. Model Two is to deploy the AFTR in the core of network to cover a large region.

When the operator consider where to deploy the AFTR, they must make trade-offs. AFTR in Model One serves few B4 elements, thus, it requires less powerful AFTR. Moreover, the traffic flows are more evenly distributed to the AFTRs. However, it requires to deploy more AFTRs to cover the entire network. Often the operation cost increases proportionally to the number of network equipment. AFTR in Model Two covers larger area, thus, it serves more B4 elements. The operator could deploy only few AFTRs in the strategic locations to support the entire subscriber base. However, this model requires more powerful AFTR to sustain the load at peak hours. Since the AFTR would support B4 elements from different regions, the AFTR would be deployed deeper in the network and steer more traffic flows to the network where the AFTR is located.

DS-Lite framework can be incrementally deployed. An operator may consider to start with Model Two. When the demand increases, they could push the AFTR closer to the edge which would effectively become Model One.

2.11. AFTR Considerations for Geographically Aware Services

By centralizing public IPv4 addresses, each address no longer represents a single machine, a single household, or a single small office. The address now represents hundreds of machines, homes, and offices related only in that they are behind the same AFTR. Identification by IP address becomes more difficult and thus applications that assume such geographic information may not work as intended.

Various applications and services will place their servers in such a way to locate them near sets of user so that this will lessen the latency on the client end. In addition, having sufficient geographical coverage can indirectly improve end-to-end latency. An example is that nameservers typically return results optimized for the DNS resolver’s location. Deployment of AFTR could be done in such a way as not to negatively impact the geographical nature of

these services. This can be done by making sure that AFTR deployments are geographically distributed so that existing assumptions of the clients source IP address by geographically aware servers can be maintained. Another possibility the application could rely on location information such as GPS co-ordination to identify the user's location. This technique is commonly used in mobile deployment where the mobile devices are probably behind a NAT device.

2.12. Impacts on QoS

As with tunneling in general there are challenges with deep packet inspection with DS-Lite for purposes of QoS. Service Providers commonly uses DSCP to classify and prioritize different types of traffic. DS-Lite tunnel can be seen as particular case of uniform conceptual tunnel model described in section 3.1 of [RFC2983]. The uniform model views an IP tunnel as just a necessary mechanism to get traffic to its destination, but the tunnel has no significant impact on traffic conditioning. In this model, any packet has exactly one DS Field that is used for traffic conditioning at any point and it is the field in the outermost IP header. In DS-Lite model this is the Traffic Class field in IPv6 header. According to [RFC2983] implementations of this model copy the DS value to the outer IP header at encapsulation and copy the outer header's DSCP value to the inner IP header at decapsulation. Applying the described model to DS-Lite scenario, it is recommended that the AFTR propagates the DSCP value in the IPv4 header to the IPv6 header after the encapsulation for the downstream traffic and, in the same way, the B4 propagates the DSCP value in the IPv4 header to the IPv6 header after the encapsulation for the upstream traffic.

2.13. Port Forwarding Considerations

Some applications require accepting incoming UDP or TCP traffic. When the remote host is on IPv4, the incoming traffic will be directed towards an IPv4 address. Some applications use (UPnP-IGD) (e.g., XBox) or ICE [I-D.ietf-mmusic-ice] (e.g., SIP, Yahoo!, Google, Microsoft chat networks), other applications have all but completely abandoned incoming connections (e.g., most FTP transfers use passive mode). But some applications rely on ALGs, UPnP IGD, or manual port configuration. Port Control Protocol (PCP) [I-D.ietf-pcp-base] is designed to address this issues.

2.14. DS-Lite Tunnel Security

Section 11 of [I-D.ietf-softwire-dual-stack-lite] describes security issues associated to DS-Lite mechanism. One of the recommendations contained in this section, in order to limit service offered by AFTR only to registered customers, is to implement IPv6 ingress filter on

the AFTR's tunnel interface to accept only the IPv6 address range defined in the filter. This approach requires to know in advance the IPv6 prefix delegated to the customers in order to be able to configure the filter.

An alternative way to achieve the same goal and to provide some form of access control to the DS-Lite tunnel, is to use DHCPv6 Leasequery defined in [RFC5007]. When the AFTR receives a packet from an unknown (new) prefix it issues a DHCPv6 Leasequery based on IPv6 address to the DHCPv6 server in order to verify if that prefix was previously delegated by the DHCPv6 server to that specific client. The DHCPv6 Server will reply with the delegated prefix and the associated lease. If the two prefix are the same the ATFR accepts the packet otherwise it drops it and it denies the service.

2.15. IPv6-only Network considerations

In environments where the service provider wants to deploy AFTR in the IPv6 core network, the AFTR nodes may not have direct IPv4 connectivity. In this scenario the service provider extends the IPv6-only boundary to the border of the network and only the border routers have IPv4 connectivity. For both scalability and performance purposes AFTR capabilities are located in the IPv6-only core closer to B4 elements. The service provider assigns only IPv6 prefixes to the B4 capable devices but also continues to provide IPv4 services to these customers. In this scenario the AFTR has only IPv6-connectivity and must be able to send and receive IPv4 packets. Enhancements to the DS-LITE AFTR are required to achive this. [I-D.boucadair-softwire-dslite-v6only] describes such issues and enhancements to DS-Lite in IPv6-only deployments.

3. B4 Deployment Considerations

In order to configure the IPv4-in-IPv6 tunnel, the B4 element needs the IPv6 address of the AFTR element. This IPv6 address can be configured using a variety of methods, ranging from an out-of-band mechanism, manual configuration or a variety of DHCPv6 options. In order to guarantee interoperability, a B4 element should implement the DHCPv6 option defined in [I-D.ietf-softwire-ds-lite-tunnel-option]. The DHCP server must be reachable via normal DHCP request channels from the B4, and it must be configured with the AFTR address. In Broadband Access scenario where AAA/RADUIS is used for provisioning user profiles in the BNAS, [I-D.ietf-softwire-dslite-radius-ext] may be used. BNAS will learn the AFTR address from the RADIUS attribute and act as the DHCPv6 server for the B4s.

3.1. DNS deployment Considerations

[I-D.ietf-softwire-dual-stack-lite] recommends configuring the B4 with a DNS proxy resolver, which will forward queries to an external recursive resolver over IPv6. Alternately, the B4 proxy resolver can be statically configured with the IPv4 address of an external recursive resolver. In this case, DNS traffic to the external resolver will be tunneled through IPv6 to the AFTR. Note that the B4 must also be statically configured with an IPv4 address in order to source packets; the draft recommends an address in the 192.0.0.0/29 range. Even more simply, you could eliminate the DNS proxy, and configure the DHCP server on the B4 to give its clients the IPv4 address of an external recursive resolver. Because of the extra traffic through the AFTR, and because of the need to statically configure the B4, these alternate solutions are likely to be unsatisfactory in a production environment. However, they may be desirable in a testing or demonstration environment.

4. Security Considerations

This document does not present any new security issues. [I-D.ietf-softwire-dual-stack-lite] discusses DS-Lite related security issues. General NAT security issues are not repeated here.

Some of the security issues with carrier-grade NAT result directly from the sharing of the routable IPv4 address. Addresses and timestamps are often used to identify a particular user, but with shared addresses, more information (i.e., protocol and port numbers) is needed. This impacts software used for logging and tracing spam, denial of service attacks, and other abuses. Devices on the customers side may try to carry out general attacks against systems on the global Internet or against other customers by using inappropriate IPv4 source addresses inside tunneled traffic. The AFTR needs to protect against such abuse. One customer may try to carry out a denial of service attack against other customers by monopolizing the available port numbers. The AFTR needs to ensure equitable access. At a more sophisticated level, a customer may try to attack specific ports used by other customers. This may be more difficult to detect and to mitigate without a complete system for authentication by port umber, which would represent a huge security requirement.

5. Conclusion

DS-Lite provides new functionality to transition IPv4 traffic to IPv6 addresses. As the supply of unique IPv4 addresses diminishes,

service providers can now allocate new subscriber homes IPv6 addresses and IPv6-capable equipment. DS-Lite provides a means for the private IPv4 addresses behind the IPv6 equipment to reach the IPv4 network.

This document discusses the issues that arise when deploying DS-Lite in various deployment modes. Hence, this document can be a useful reference for service providers and network designers. Deployment considerations of the B4, AFTR and DNS have been discussed and recommendations for their usage have been documented.

6. Acknowledgement

TBD

7. IANA Considerations

This memo includes no request to IANA.

8. References

8.1. Normative References

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-13 (work in progress), July 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-dslite-radius-ext]
Maglione, R. and A. Durand, "RADIUS Extensions for Dual-Stack Lite", draft-ietf-softwire-dslite-radius-ext-02 (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-11 (work in progress), May 2011.

[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.

[RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng, "DHCPv6 Leasequery", RFC 5007, September 2007.

8.2. Informative References

[I-D.boucadair-intarea-nat-reveal-analysis]
Boucadair, M., Touch, J., Levis, P., and R. Penno, "Analysis of Solution Candidates to Reveal a Host Identifier in Shared Address Deployments", draft-boucadair-intarea-nat-reveal-analysis-03 (work in progress), June 2011.

[I-D.boucadair-softwire-dslite-v6only]
Boucadair, M., Jacquenet, C., Grimault, J., Kassi-Lahlou, M., Levis, P., Cheng, D., and Y. Lee, "Deploying Dual-Stack Lite in IPv6 Network", draft-boucadair-softwire-dslite-v6only-01 (work in progress), April 2011.

[I-D.ietf-intarea-server-logging-recommendations]
Durand, A., Gashinsky, I., Lee, D., and S. Sheppard, "Logging recommendations for Internet facing servers", draft-ietf-intarea-server-logging-recommendations-04 (work in progress), April 2011.

[I-D.ietf-intarea-shared-addressing-issues]
Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", draft-ietf-intarea-shared-addressing-issues-05 (work in progress), March 2011.

[I-D.ietf-mmusic-ice]
Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-19 (work in progress), October 2007.

[I-D.ietf-v6ops-ipv6-cpe-router]
Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", draft-ietf-v6ops-ipv6-cpe-router-09 (work in progress), December 2010.

[I-D.xu-behave-stateful-nat-standby]
Xu, X., Boucadair, M., Lee, Y., and G. Chen, "Redundancy Requirements and Framework for Stateful Network Address Translators (NAT)", draft-xu-behave-stateful-nat-standby-06 (work in progress), October 2010.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, January 2010.

Authors' Addresses

Yiu L. Lee
Comcast
One Comcast Center
Philadelphia, PA 19103
U.S.A.

Email: yiu_lee@cable.comcast.com
URI: http://www.comcast.com

Roberta Maglione
Telecom Italia
Via Reiss Romoli 274
Torino 10148
Italy

Email: roberta.maglione@telecomitalia.it
URI:

Carl Williams
MCSR Labs
Philadelphia
U.S.A.

Email: carlw@mcsr-labs.org

Christian Jacquenet
France Telecom
Rennes
France

Email: christian.jacquenet@orange-ftgroup.com

Mohamed Boucadair
France Telecom
Rennes
France

Email: mohamed.boucadair@orange-ftgroup.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 4, 2012

N. Matsuhira
Fujitsu Limited

July 3, 2011

Motivation for developing Stateless Automatic IPv4 over IPv6 Tunneling (SA46T)
draft-matsuhira-sa46t-motivation-00

Abstract

This document describe a motivation for developing IPv4 over IPv6 Tunneling solution from standing position of Stateless Autimatic IPv4 over IPv6 Tunneling (SA46T) and SA46T with address sharing (SA46T-AS).

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 RFC 2119 [RFC2119].

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 4, 2012.

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

This document describe a motivation for developing IPv4 over IPv6 Tunneling solution from standing position of Stateless Automatic IPv4 over IPv6 Tunneling (SA46T)[I-D.draft-matsuhira-sa46t-spec] and SA46T with address sharing (SA46T-AS)[I-D.draft-matsuhira-sa46t-as].

2. Recongition of IPv6 Transtion stage

2.1. Stages of IPv6 Transition

There is an idea that divited the transiton from IPv4 to IPv6 into three stages, early stage, middle stage and end stage. In early stage, majority of the Internet are based on IPv4, so IPv6 over IPv4 tunneling technologies are considerd to be useful. In middle stage, majority of the Internet are based on Dual Stack (Both IPv4 and IPv6), so both IPv4 and IPv6 will treat as is, and no major tunneling technologies are considerd to be use. In end stage, majority of the Internet are based on IPv6, so IPv4 over IPv6 tunneling technologies are considerd to may be useful as option, because dual stack based operation still effective in end stage.

It seems that a lot of people should have thought that the majority of transiton to IPv6 are completed before the IPv4 address exhaustion. In this recognition, IPv4 over IPv6 tunneling solution is not indispensable, but is some operational option, exclude artifical made IPv6 only network.

2.2. IPv4 address exhaution

The IPv4 address exhaution already became the real.

In 03-Feb-2011, IANA Unallocated Address Pool was exhausted. And in 19-Apr-2011, APNIC unallocated address pool was exhaused. Other RIRs, unallocated address pool does not exhaused now, however, it should be a matter of time. For more details, please refer to IPv4 address report , http://www.potaroo.net/tools/ipv4/.

The IPv4 address exhaution has already become the reality. That mean that the environment of IPv6 only also becomes the reality, too.

2.3. Current stage of IPv6 Transition

When paying attention to IPv6 traffic, current stage of IPv6 transiton should be very very early stage, however IPv4 address exhaution is the reality.

It should be recognized that a big gap is caused between the situation of IPv6 deployment and the situation of IPv4 address supply.

IPv4 is still majority now, and there are few IPv6 environment except research networks. It is not easy to change from IPv4 to IPv6 suddenly especially servers or services. That mean, IPv4 address still required for continuance of current IPv4 service with necessary minimum enhancing.

3. Motivation of developing SA46T and SA46T-AS

The IPv4 traffic is generated by the IPv4 host. On the other hand, in general, to carry the IPv4 traffic, the IPv4 routing function is necessary. However, if the IPv4 over IPv6 tunneling technology is used, it is enough by the IPv6 routing function.

Following are the motivation of developing SA46T.

- Develop simple and scalable IPv4 over IPv6 tunneling technology.
- Enabele single stack operaton by IPv6 in the backbone network.
- Can collect the IPv4 global address from where it is not indispensable, and reallocate the IPv4 global address to where it is indispensable.
- Can still use IPv4 address (both global and private) with access environments is IPv6 only.
- Support IPv4 address reuse and IPv4 address sharing if necessary.
- Can deploy to IPv6 in stub network with their own peace

4. Designe goal

4.1. Can install into existing network

The IPv4 address can be collected only from an existing network where the IPv4 address is used. Therefore, it is necessary to be able to install it into an existing network.

Of cource, It is possible to use it even on a new network.

4.2. Less tunnel configuration

In a existing tunnel technique, the configuration of N^2 piece is needed for number N of tunnels end points connecting for full mesh topology. When N is small, it is not a problem, however when N is large, many many configuration required, then reality disappear. It cannot be considered the technology with the scalability.

The achievement of the scalability is require for really use from small network to large network. This mean technolohy require less configuration.

4.3. Simple install strategy

In general, the tunnel technique is a technology that makes a virtual link between two arbitrary interfaces. Flexibility is very high. However, such flexibility may causes the recursive tunnelung (tunnel in tunnel), and cause the difficulties for management and the trouble shooting.

It is thought that this flexiblity make difficultly for large-scale development. That mean simple install strategy is required for avoiding such problems.

4.4. Can treat both IPv4 Global and IPv4 Privates

For applying backbone networks, it should treat stub network which used not only IPv4 global address but also IPv4 private address. Moreover, it should treat many networks which use IPv4 private address. It means it is unaffected in the reused address, or non gloablly unique addess.

Moreover, it should not depend with the range of IP address. That mean it should be no dependence with the addresses used in stub networks.

4.5. Can install into varius networks

It is preferable to be able to apply widely.

For example, it should apply access network, backbone network, data-center network, enterprise network, etc. Moreover, It has no dependency with Layer two technology, such as wire and wireless.

5. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

Security consideration does not discussed in this memo.

7. Acknowledgements

SA46T implementation was tested in Fujitsu, WIDE camp network in Septemper 2010, and NICT JGN2Plus testbed in February 2011. And SA46T was demonstrated at Interop 2011 Tokyo in June 2011.

The author would like to thank all the people who assist, support and help above tests and demonstration, especially WIDE camp network team, NICT JGN2Plus / JGN-X team, Interop Shownet NOC team and in Fujitsu.

8. References

8.1. Normative References

[I-D.draft-matsuhira-sa46t-as]
Matsuhira, N., "Stateless Automatic IPv4 over IPv6 Tunneling with IPv4 Address Sharing", April 2011.

[I-D.draft-matsuhira-sa46t-spec]
Matsuhira, N., "Stateless Automatic IPv4 over IPv6 Tunneling: Specification", July 2011.

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

8.2. Informative References

Author’s Address

Naoki Matsuhira
Fujitsu Limited
17-25, Shinkamata 1-chome, Ota-ku
Tokyo, 144-8588
Japan

Phone: +81-3-6424-6270
Fax:
Email: matsuhira@jp.fujitsu.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2012

T. Murakami
IP Infusion
O. Troan
cisco

July 4, 2011

IPv4 Residual Deployment on IPv6 infrastructure - protocol specification
draft-murakami-softwire-4rd-00

Abstract

This document specifies an automatic tunneling mechanism for providing IPv4 connectivity service to end users over a service provider's IPv6 network. Key aspects include stateless operation, sharing of IPv4 addresses, and an algorithmic mapping between IPv4 addresses and IPv6 tunnel endpoints.

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 5, 2012.

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

Table of Contents

1. Introduction

4rd is a protocol mechanism to deploy IPv4 to sites via a service provider's (SP's) IPv6 network. Similar to Dual-Stack Lite [I-D.ietf-softwire-dual-stack-lite], 4rd is designed to allow IPv4 traffic to be delivered over an IPv6 network without the direct provisioning of IPv4 addresses. 4rd can provide an IPv4 prefix, an IPv4 address or a shared IPv4 address. Like 6rd [RFC5969], 4rd is operated in a fully stateless manner within the SP network. The motivation for a stateless alternative to Dual-Stack Lite is described in "Motivations for Stateless IPv4 over IPv6 Migration Solutions" [I-D.operators-softwire-stateless-4v6-motivation].

4rd relies on IPv6 and is designed to deliver production-quality dual-stack service while allowing IPv4 to be phased out within the SP network. The phasing out of IPv4 within the SP network is independent of whether the end user disables IPv4 service or not. Further, "Greenfield" IPv6-only networks may use 4rd in order to deliver IPv4 to sites via the IPv6 network in a way that does not require protocol translation between IPv4 and IPv6.

4rd utilizes an algorithmic mapping between the IPv6 and IPv4 addresses that are assigned for use within the SP network. This mapping provides automatic determination of IPv6 tunnel endpoints from IPv4 destination addresses, allowing the stateless operation of 4rd. 4rd views the IPv6 network as a link layer for IPv4 and supports an automatic tunneling abstraction similar to the Non-Broadcast Multiple Access (NBMA) [RFC2491] model.

The 4rd algorithmic mapping is also used to automatically provision IPv4 addresses and allocating a set of non-overlapping ports for each 4rd CE. The "SP-facing" (i.e., "WAN") side of the 4rd CE, operate as native IPv6 interface with no need for IPv4 operation or support. On the "end-user-facing" (i.e., "LAN") side of a CE, IPv6 and IPv4 are implemented as for any native dual-stack service delivered by the SP.

A 4rd domain consists of 4rd Customer Edge (CE) routers and one or more 4rd Border Relays (BRs). IPv4 packets encapsulated by 4rd follow the IPv6 routing topology within the SP network between CEs and among CEs and BRs. CE to CE traffic is direct, while BRs are traversed only for IPv4 packets that are destined to or are arriving from outside a given 4rd domain. As 4rd is stateless, BRs may be reached using anycast for failover and resiliency.

4rd does not require any stateful NAPT [RFC3022] functions at the BRs or elsewhere within the SP network. Instead, 4rd allows for sharing of IPv4 addresses among multiple sites by automatically allocating a set of non-overlapping ports for each CE as part of the stateless

mapping function. It is expected that the CE will, in turn, perform local IPv4 Network Address and Port Translation (NAPT) [RFC3022] functions for the site as is commonly performed today, except avoiding ports outside of the allocated port set. Although 4rd is designed primarily to support IPv4 deployment to a customer site (such as a residential home network) by an SP, it can equally be applied to an individual host acting as a CE router.

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 RFC 2119 [RFC2119].

3. Terminology

4rd domain (Domain): A set of 4rd CEs and BRs connected to the same virtual 4rd link. A service provider may deploy 4rd with a single 4rd domain, or may utilize multiple 4rd domains. Each domain requires a separate 4rd prefix.

4rd Border Relay (BR): A 4rd-enabled router managed by the service provider at the edge of a 4rd domain. A Border Relay router has at least one of each of the following: an IPv6-enabled interface, a 4rd virtual interface acting as an endpoint for the 4rd IPv4 in IPv6 tunnel, and an IPv4 interface connected to the native IPv4 network. A 4rd BR may also be referred to simply as a "BR" within the context of 4rd.

4rd Customer Edge (CE): A device functioning as a Customer Edge router in a 4rd deployment. In a residential broadband deployment, this type of device is sometimes referred to as a "Residential Gateway" (RG) or "Customer Premises Equipment" (CPE). A typical 4rd CE serving a residential site has one WAN side interface, one or more LAN side interfaces, and a 4rd virtual interface. A 4rd CE may also be referred to simply as a "CE" within the context of 4rd.

CE IPv6 prefix: The IPv6 prefix assigned to a CE by other means than 4rd itself, and used by 4rd to derive a CE 4rd prefix.

CE IPv6 address: The IPv6 address given to the CE as part of normal IPv6 Internet access. This address is used by a 4rd CE to create the 4rd prefix as well as to send and receive IPv6-encapsulated IPv4 packets.

CE 4rd prefix: The 4rd prefix of the CE. It is derived from the CE IPv6 prefix by a mapping rule according to Section 5.1. Depending on its length, it is an IPv4 prefix, an IPv4 address, or a shared IPv4 address followed by a Port-set ID (Section 5.1.2).

Port-set ID: In a CE 4rd prefix longer than 32 bits, bits that follow the first 32. It algorithmically identifies a set of ports exclusively assigned to the CE. As specified in Section Section 5.1.2, the set can comprise up to 4 disjoint port ranges.

Domain IPv6 prefix: An IPv6 prefix assigned by an ISP to a 4rd domain.

Domain IPv4 prefix: A 4rd prefix assigned by an ISP to the 4rd domain.

IPv4 Embedded Address (EA) bits: The IPv4 EA-bits in the IPv6 address identify an IPv4 prefix, IPv4 address or part of IPv4 address and port set.

Shared IPv4 address: An IPv4 address that is shared among multiple nodes. Each node has a separate part of the transport layer port space.

4. 4rd Configuration

The IPv4 prefix, IPv4 address or shared IPv4 address for use at a customer site is created by extracting the IPv4 embedded address (EA-bits) from the IPv6 prefix delegated to the site. Combined with the 4rd IPv4 prefix, the IPv4 prefix, IPv4 address or shared IPv4 address is automatically created by the CE for the customer site when IPv6 service is obtained.

For a given 4rd domain, the BR and CE MUST be configured with a set of mapping rules and BR IPv6 addresses. The configured values for these elements MUST be identical for all CEs and BRs within a given 4rd domain.

A mapping rule consist of the following elements: a Domain IPv6 prefix and prefix length, a Domain 4rd prefix and prefix length, CE IPv6 Prefix length, and a Domain IPv6 suffix and length. See section (Section 5.1) for a detailed description of mapping rules.

4.1. Customer Edge Configuration

The 4rd configuration elements are set to values that are the same across all CEs within a 4rd domain. The values may be configured in a variety of manners, including provisioning methods such as the Broadband Forum's "TR-69" [TR069] Residential Gateway management interface, an XML-based object retrieved after IPv6 connectivity is established, a DNS record, an SMIv2 MIB [RFC2578], or manual configuration by an administrator. A companion document [I-D.mrugalski-dhc-dhcpv6-4rd]describes how to configure the necessary parameters via IPv6 DHCP. A CE that allows IPv6 configuration by IPv6 DHCP SHOULD implement this option. Other configuration and management methods may use the format described by this option for consistency and convenience of implementation on CEs that support multiple configuration methods.

The only remaining provisioning information the CE requires in order to calculate the 4rd address and enable IPv6 connectivity is an IPv6 prefix for the CE. This CE IPv6 prefix is configured as part of obtaining IPv6 Internet access (i.e., configured via SLAAC, DHCPv6, DHCPv6 PD, or otherwise).

A single 4rd CE MAY be connected to more than one 4rd domain. Each domain a given CE operates within would require its own set of 4rd configuration elements and would generate its own 4rd address.

5. Algorithmic mapping

5.1. Mapping Rules

5.1.1. From a CE IPv6 Prefix to a CE 4rd Prefix

A 4rd mapping rule establishes a 1:1 mapping between CE IPv6 prefixes and CE 4rd prefixes.

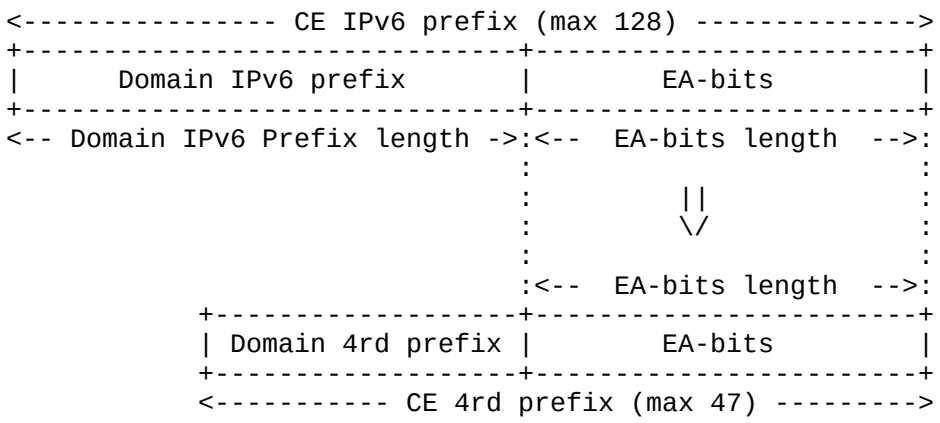


Figure 1: From a CE IPv6 Prefix to a CE 4rd Prefix

A CE derives its CE 4rd prefix from the CE IPv6 prefix, using parameters of the applicable mapping rule. If the domain has several mapping rules, the rule that applies is that whose Domain IPv6 prefix has the longest match with the CE IPv6 prefix. As shown in Figure 1, the CE 4rd prefix is created by concatenating the Domain 4rd prefix with the IPv4 EA-bits, where the IPv4 EA-bits is the remainder of the CE IPv6 prefix after the Domain IPv6 prefix (the length of the Domain IPv6 prefix is defined by the mapping rule).

5.1.2. From a CE 4rd Prefix to a Port-set ID

Depending on its length, a CE 4rd prefix is either an IPv4 prefix, a full IPv4 address, or a shared IPv4 address followed by a Port-set ID (Figure 2). If it includes a port set ID, this ID specifies which ports are assigned to the the CE for its exclusive use (Section 5.1.3).

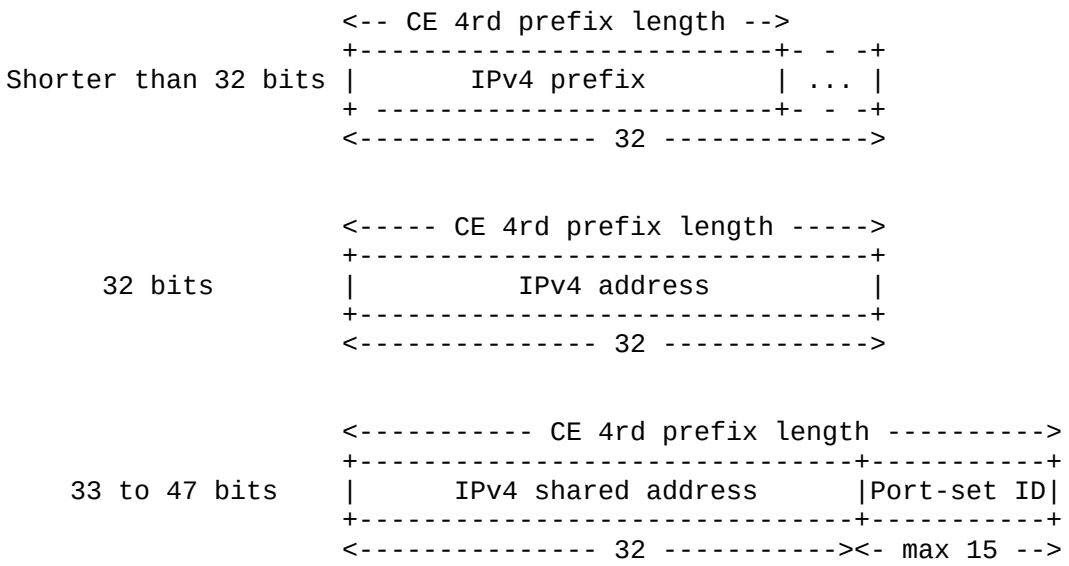


Figure 2: Variants of CE 4rd prefixes

5.1.3. From a Port-Set ID to a Port Set

The value of a Port-set ID specifies which ports can be used by a transport layer protocol (UDP, TCP, SCTP etc). Design constraint of the algorithm are the following:

Fairness with respect to special-value ports: No port-set must contain any well-known ports [IANA reference].

Fairness with respect to the number of ports For a Port-set-ID's having the same length, all sets must have the same number of ports.

Exhaustiveness For any Port-set-ID length, the aggregate of port sets assigned for all values must include all ordinary-value ports.

If the Port-set ID has 1 to 12 bits, the set comprises 4 port ranges. As shown in Figure 3, each port range is defined by its port prefix, made of a range-specific "head" followed by the Port-set ID. Head values are in binary 1, 01, 001, and 0001. They are chosen to exclude ports 0-4095 and only them.

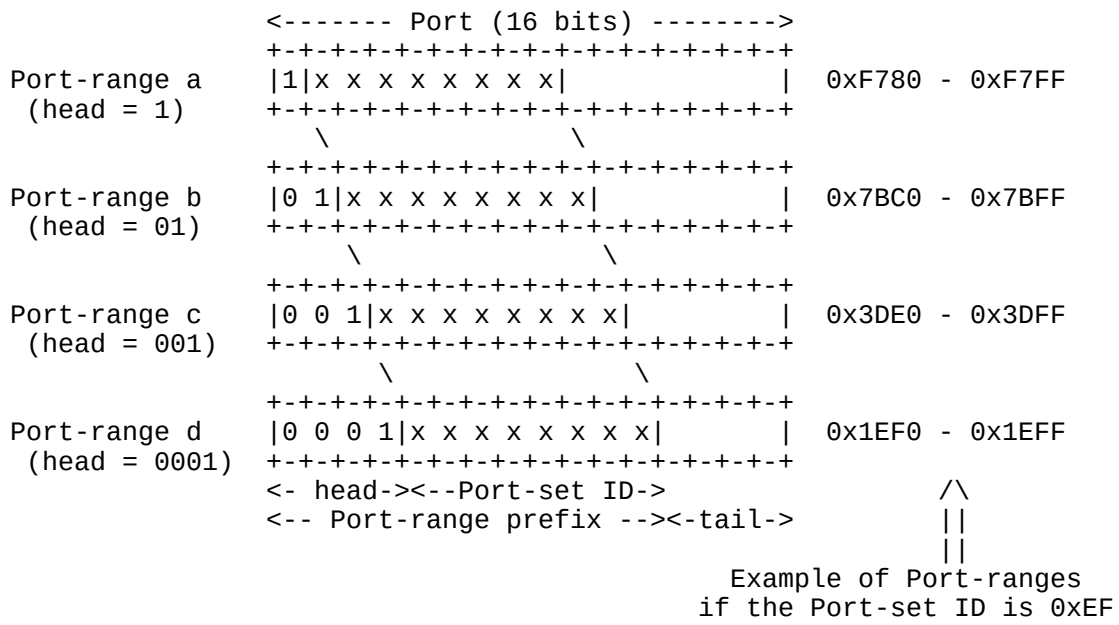


Figure 3: From Port-set ID to Port ranges

In the Port-set ID has 13 bits, only the 3 port ranges are assigned, having heads 1, 01, and 001. If it has 14 bits, only the 2 port ranges having heads 1 and 01 are assigned. If it has 15 bits, only the port range having head 1 is assigned. (In these three cases, the smallest port range has only one element).

5.1.4. From an IPv4 Address or IPv4 Address + Port to a CE IPv6 Address

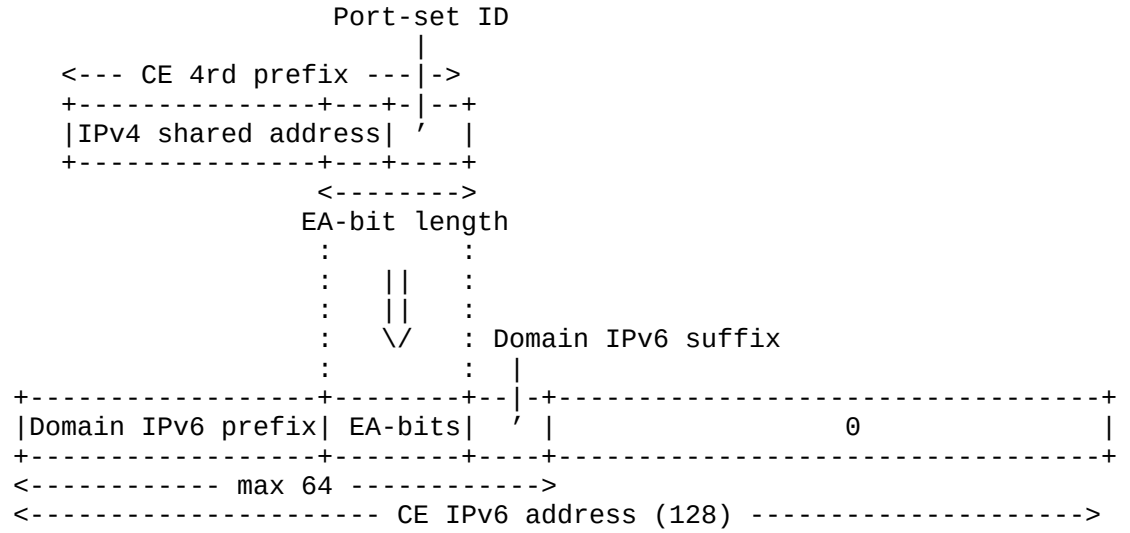


Figure 4: From 4rd Prefix to IPv6 address (shared IPv4 address case)

In order to find whether a CE IPv6 address can be derived from an IPv4 address, or an IPv4 address + a port, a mapping rule has to be found that matches the IPv4 information:

- If a mapping rule has a length L of CE IPv4 prefixes which does not exceed 32 bits, there is a match if the IPv4 address starts with the Domain 4rd prefix. The CE 4rd prefix is then the first L bits of the IPv4 address.

- If a mapping rule has a length L of CE IPv4 prefixes which exceeds 32 bits, the match can only be found with the IPv4 address and the port. For this, the port is examined to determine which port-range head it starts with: 1, 01,001, or 0001. The N bits that follow this head are taken as Port-set ID, where N is the length of Port set ID of the mapping rule. The CE 4rd prefix is then made of the IPv4 address followed by the Port-set ID.

If a match has been found, the CE IPv6 prefix is then made of the Domain IPv6 prefix followed by bits of the CE 4rd prefix that follow the Domain 4rd prefix, followed by the Domain IPv6 prefix of the mapping rule if there is one, and followed by 0's up to 128 bits to make a complete IPv6 address ([RFC4291]. Figure 4 illustrates this process in the case of a shared IPv4 address.

6. Encapsulation and Fragmentation Consideration

Maximum transmission unit (MTU) and fragmentation issues for IPv4 in IPv6 tunneling are discussed in detail in Section 7.2 of [RFC2473]. 4rd's scope is limited to a service provider network. IPv6 Path MTU discovery MAY be used to adjust the MTU of the tunnel as described in Section 7.2 of [RFC2473], or the 4rd Tunnel MTU might be explicitly configured.

The use of an anycast source address could lead to any ICMP error message generated on the path being sent to a different BR. Therefore, using dynamic tunnel MTU Section 7.2 of [RFC2473] is subject to IPv6 Path MTU blackholes.

Multiple BRs using the same anycast source address could send fragmented packets to the same 4rd CE at the same time. If the fragmented packets from different BRs happen to use the same fragment ID, incorrect reassembly might occur. For this reason, a BR using an anycast source address MUST NOT fragment the IPv6 encapsulated packet.

If the MTU is well-managed such that the IPv6 MTU on the CE WAN side interface is set so that no fragmentation occurs within the boundary of the SP, then the 4rd Tunnel MTU should be set to the known IPv6 MTU minus the size of the encapsulating IPv6 header (40 bytes). For example, if the IPv6 MTU is known to be 1500 bytes, the 4rd Tunnel MTU might be set to 1460 bytes. Absent more specific information, the 4rd Tunnel MTU SHOULD default to 1280 bytes.

For 4rd domain traversal, IPv4 packets are encapsulated in IPv6 packets whose Next header is set to 4 (i.e. IPv4). If fragmentation of IPv6 packets is needed, it is performed according to [RFC2460]. Absent more specific information, the path MTU of a 4rd Domain has to be set to 1280 [RFC2460].

In domains where IPv4 addresses are not shared, IPv6 destinations are derived from IPv4 addresses alone. Thus, each IPv4 packet can be encapsulated and decapsulated independently of each other. 4rd processing is completely stateless.

On the other hand, in domains where IPv4 addresses are shared, BR's and CE's can have to encapsulate IPv4 packets whose IPv6 destinations depend on destination ports. Precautions are needed, due to the fact that the destination port of a fragmented datagram is available only in its first fragment. A sufficient precaution consists in reassembling each datagram received in multiple packets, and to treat it as though it would have been received in single packet. This function is such that 4rd is in this case stateful at the IP layer.

(This is common with DS-lite and NAT64/DNS64 which, in addition, are stateful at the transport layer.) At Domain entrance, this ensures that all pieces of all received IPv4 datagrams go to the right IPv6 destinations.

Another peculiarity of shared IPv4 addresses is that, without precaution, a destination could simultaneously receive from different sources fragmented datagrams that have the same Datagram ID (the Identification field of [RFC0791]. This would disturb the reassembly process. To eliminate this risk, CE MUST rewrite the datagram ID to an unique value among CEs having same shared IPv4 address upon sending the packets over 4rd tunnel. This value SHOULD be generated locally within the port-range assigned to a given CE. Note that replacing a Datagram ID in an IPv4 header implies an update of its Header-checksum fieald, by adding to it the one's complement difference between the old and the new values.

7. BR and CE behaviors

(a) BR reception of an IPv4 packet

Step 1	BR looks up an appropriate mapping rule with a specific Domain 4rd prefix which has the longest match with an IPv4 destination address in the received IPv4 packet. If the mapping rule is not found, the received packet should be discarded. If the length of CE 4rd prefix associated with the mapping rule does not exceed 32 bits, BR proceeds to step 2. If the length of CE 4rd prefix exceeds 32 bits, BR checks that the received packet contains a complete IPv4 datagram. If the packet is fragmented, BR should reassemble the packet. Once BR can obtain the complete IPv4 datagram, BR proceeds to step 2 as though the datagram has been received in a single packet.
Step 2	BR generates a CE IPv6 address from the IPv4 destination address or the IPv4 destination address and the destination port based on the mapping rule found in step 1. If the CE IPv6 address can be successfully generated, BR encapsulates the IPv4 packet in IPv6 and forwards the IPv6 packet via the IPv6 interface. If the length of the IPv6 encapsulated packet exceeds the MTU of the IPv6 interface, the fragmentation should be done in

IPv6.

(b) BR reception of an IPv6 packet

Step 1 If the received IPv6 packet is fragmented, the reassembly should be done in IPv6 at first. Once BR obtains a complete IPv6 packet, BR looks up an appropriate mapping rule with a specific Domain 4rd prefix which has the longest match with an IPv4 source address in the encapsulated IPv4 packet. If the mapping rule is not found, the received IPv6 packet should be discarded. BR derives a CE IPv6 address from the IPv4 source address or the IPv4 source address and the source port in the encapsulated IPv4 packet based on the mapping rule. If the CE IPv6 address is eqaul to the IPv6 source address in the received IPv6 packet, BR decapsulates the IPv4 packet and then forward it via the IPv4 interface.

(c) CE reception of an IPv4 packet

Step 1 CE looks up an appropriate mapping rule with a specific Doamin 4rd prefix which has the longest match with an IPv4 destination address in the received IPv4 packet. If the mapping rule is found, the CE 4rd prefix must be checked. If the length does not exceeds 32 bits, CE proceeds to step 2. If the length exceeds 32 bits, CE checks that the received IPv4 packet contains a complete IPv4 datagram. If the packet is fragmented, CE should reassemble the packet. Once CE can obtain the complete IPv4 datagram, CE proceeds to step 2 as though the datagram has been received in a single packet. If the mapping rule is not found, CE proceeds to step 2.

Step 2 If the mapping rule is found in step 1, CE derives a IPv6 destination address from the IPv4 destination address or the IPv4 destination address and the destination port based on the mapping rule. If the IPv6 destination address can be derived successfully, CE encapsulates the IPv4 packet in IPv6 whose destination address is set to the derived IPv6 address. If the mapping rule is

not found in step 1, CE encapsulates the IPv4 packet in IPv6 whose destination address is set to BR IPv6 address. Then CE forwards the IPv6 packet via IPv6 interface. If the length of the IPv6 packet exceeds the MTU of the IPv6 interface, the fragmentation should be done in IPv6. Moreover, if using IPv4 shared address, a Datagram ID in the received IPv4 header must be over-written before encapsulating the IPv4 packet in IPv6. In case of shared IPv4 address, the Datagram ID must be unique among CEs sharing the same IPv4 address. Hence, CE should assign the unique value and set this value to the datagram ID in IPv4 header. This value may be generated from the port-range assigned to the CE to keep the uniqueness among CEs sharing same IPv4 address.

(d) CE reception of an IPv6 packet

Step 1 If the received IPv6 packet is fragmented, the reassembly should be done in IPv6 at first. Once CE obtains a complete IPv6 packet, CE looks up an appropriate mapping rule with s specific Domain 4rd prefix which has the longest match with an IPv4 source address in the encapsulated IPv4 packet. If the mapping rule is found, CE derives a CE IPv6 address from the IPv4 source address or the IPv4 source address and the source port based on the mapping rule and then checks that the IPv6 source address of the received IPv6 packet is matched to it. If the mapping rule is not found, CE checks that the IPv6 source address is matched to BR IPv6 address. In case of success, CE decapsulates the IPv4 packet and forward it via the IPv4 interface.

8. NAT considerations

NAT44 should be implemented in CPE which has 4rd CE function. The NAT44 must conform that best current practice documented in [RFC4787], [RFC5508] and [RFC5382]. When there are restricted available port numbers in a given 4rd CE described in Section 5.1.3, the NAT44 must restrict mapping ports within the port-set.

9. ICMP

ICMP message should be supported in 4rd domain. Hence, the NAT44 in 4rd CE must implement the behavior for ICMP message conforming to the best current practice documented in [RFC5508].

If a 4rd CE receives an ICMP message having ICMP identifier field in ICMP header, NAT44 in the 4rd CE must rewrite this field to a specific value assigned from the port-set described in Section 5.1.3. BR and other CEs must handle this field similar to the port number in tcp/udp header upon receiving the ICMP message with ICMP identifier field.

If a 4rd BR and CE receives an ICMP error message without ICMP identifier field for some errors that is detected inside a IPv6 tunnel, a 4rd BR and CE should replay the ICMP error message to the original source. This behavior should be implemented conforming to the section 8 of [RFC2473]. The 4rd BR and CE obtain the origianl IPv6 tunnel packet storing in ICMP payload and then decapsulate IPv4 packet. Finally the 4rd BR and CE generate a new ICMP error message from the decapsulated IPv4 packet and then forward it.

If a 4rd BR receives an ICMP error message on its IPv4 interface, the 4rd BR should replay the ICMP message to an appropriate 4rd CE. If IPv4 address is not shared, the 4rd BR generates a CE IPv6 address from the IPv4 destination address in the ICMP error message and encapsulates the ICMP message in IPv6. If IPv4 address is shared, the 4rd BR derives an original IPv4 packet from the ICMP payload and generates a CE IPv6 address from the source address and the source port in the original IPv4 packet. If the 4rd BR can generate the CE IPv6 address, the 4rd BR encapsulates the ICMP error message in IPv6 and then forward it to its IPv6 interface.

10. Security Considerations

Spoofing attacks: With consistency checks between IPv4 and IPv6 sources that are performed on IPv4/IPv6 packets received by BR’s and CE’s (Section 7), 4rd does not introduce any opportunity for spoofing attack that would not pre-exist in IPv6.

Denial-of-service attacks: In 4rd domains where IPv4 addresses are shared, the fact that IPv4 datagram reassembly may be necessary introduces an opportunity for DOS attacks (Section 4.4). This is inherent to address sharing, and is common with other address sharing approaches such as DS- lite and

NAT64/DNS64. The best protection against such attacks is to accelerate IPv6 enablement in both clients and servers so that, where 4rd is supported, it is less and less used.

Routing-loop attacks: This attack may exist in some automatic-tunneling scenarios are documented in [I-D.ietf-v6ops-tunnel-loops]. They cannot exist with 4rd because each BRs checks that the IPv6 source address of a received IPv6 packet is a CE address Section 5.1.

Attacks facilitated by restricted port set: From hosts that are not subject to ingress filtering of [RFC2827], some attacks are possible by intervening with faked packets during ongoing transport connections ([RFC4953], [RFC5961], [RFC6056]. The attacks depend on guessing which ports are currently used by target hosts. Using unrestricted port set which mean that are IPv6 is exactly preferable. To avoid this attacks using restricted port set, NAT44 filtering behavior must be "Address-Dependent Filtering".

11. IANA Consideration

This document makes no request of IANA.

12. Acknowledgements

This draft is based on original idea described in [I-D.despres-softwire-sam]. The authors would like to thank Remi Despres, Mark Townsley, Wojciech Dec, Olivier Vautrin and Satoru Matsushima.

13. References

13.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2491] Armitage, G., Schulter, P., Jork, M., and G. Harter, "IPv6 over Non-Broadcast Multiple Access (NBMA) networks", RFC 2491, January 1999.

[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.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

13.2. Informative References

[I-D.despres-softwire-sam]
Despres, R., "Stateless Address Mapping (SAM) - a Simplified Mesh-Softwire Model", draft-despres-softwire-sam-01 (work in progress), July 2010.

[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-11 (work in progress), May 2011.

[I-D.ietf-v6ops-tunnel-loops]
Nakibly, G. and F. Templin, "Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", draft-ietf-v6ops-tunnel-loops-07 (work in progress), May 2011.

[I-D.mrugalski-dhc-dhcpv6-4rd]
Mrugalski, T., "DHCPv6 Options for IPv4 Residual Deployment (4rd)", draft-mrugalski-dhc-dhcpv6-4rd-00 (work in progress), July 2011.

[I-D.operators-softwire-stateless-4v6-motivation]
Boucadair, M., Matsushima, S., Lee, Y., Bonness, O., Borges, I., and G. Chen, "Motivations for Stateless IPv4 over IPv6 Migration Solutions", draft-operators-softwire-stateless-4v6-motivation-02 (work in progress), June 2011.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

[RFC4953] Touch, J., "Defending TCP Against Spoofing Attacks", RFC 4953, July 2007.

[RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.

[RFC5508] Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT Behavioral Requirements for ICMP", BCP 148, RFC 5508, April 2009.

[RFC5961] Ramaiah, A., Stewart, R., and M. Dalal, "Improving TCP's Robustness to Blind In-Window Attacks", RFC 5961, August 2010.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, January 2011.

Authors' Addresses

Tetsuya Murakami
IP Infusion
1188 East Arques Avenue
Sunnyvale
USA

Email: tetsuya@ipinfusion.com

Ole Troan
cisco
Oslo
Norway

Email: ot@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2012

T. Murakami, Ed.
IP Infusion
G. Chen
H. Deng
China Mobile
W. Dec
Cisco Systems
S. Matsushima
SoftBank Telecom
July 4, 2011

4via6 Stateless Translation
draft-murakami-softwire-4v6-translation-00

Abstract

This document specify 4via6, a solution for IPv4 connectivity across IPv6 network utilizes 4rd algorithmic address mapping rule as a series of stateless IPv4 over IPv6 migration solutions. 4via6 employ stateless address translation techniques. It is useful for operators who want to provide IPv4 connectivity across restricted bandwidth IPv6 network with stateless operation.

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 5, 2012.

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

4via6 is a solution utilizes the same algorithmic address mapping rule between IPv4 addresses and IPv6 addresses defined in 4rd [I-D.murakami-softwire-4rd]. 4via6 employ stateless address translation techniques well specified in [RFC6145] with the mapping rule in order to communicate IPv4 islands across IPv6 network, instead of IPv6 encapsulation mechanism in 4rd. Address mapping rule defined in [RFC6052] is also employed to preserve correspondant address of outside 4via6 domain.

Since additional IP header is required and the size of the packet is increasing in encapsulation solutions, limited bandwidth resource in a network would be consumed by un-negligible overhead. It is undesirable in that has that limitation like wireless network. 4via6 is useful for operators who want to provide IPv4 connectivity across restricted bandwidth IPv6 network with stateless operation described in [I-D.operators-softwire-stateless-4v6-motivation].

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 RFC 2119 [RFC2119].

3. Terminology

4via6 domain (Domain): A set of 4via6 CEs and BRs connected to the same virtual link. A service provider may deploy 4via6 with a single 4via6 domain, or may utilize multiple 4via6 domains. Each domain requires a separate 4via6 prefix.

4via6 Border Relay (BR): A 4via6-enabled router managed by the service provider at the edge of a 4via6 domain. A Border Relay router has at least an IPv6-enabled interface and an IPv4 interface connected to the native IPv4 network. A 4via6 BR may also be referred to simply as a "BR" within the context of 4via6.

4via6 Customer Edge (CE): A device functioning as a Customer Edge router in a 4via6 deployment. In a residential broadband deployment, this type of device is sometimes referred to as a "Residential Gateway" (RG) or "Customer Premises Equipment"

(CPE). A typical 4via6 CE adopting 4rd rules will serve a residential site with one WAN side interface, one or more LAN side interfaces. A 4via6 CE may also be referred to simply as a "CE" within the context of 4via6.

Shared IPv4 address: An IPv4 address that is shared among multiple nodes. Each node has a separate part of the transport layer port space.

4. 4via6 Translation Framework

Figure 1 depicts the overall architecture with IPv4 users networks connected through routed IPv6 networks. Therein, IPv4 users are connected to IPv6 network via CPE with 4via6 translation modules.

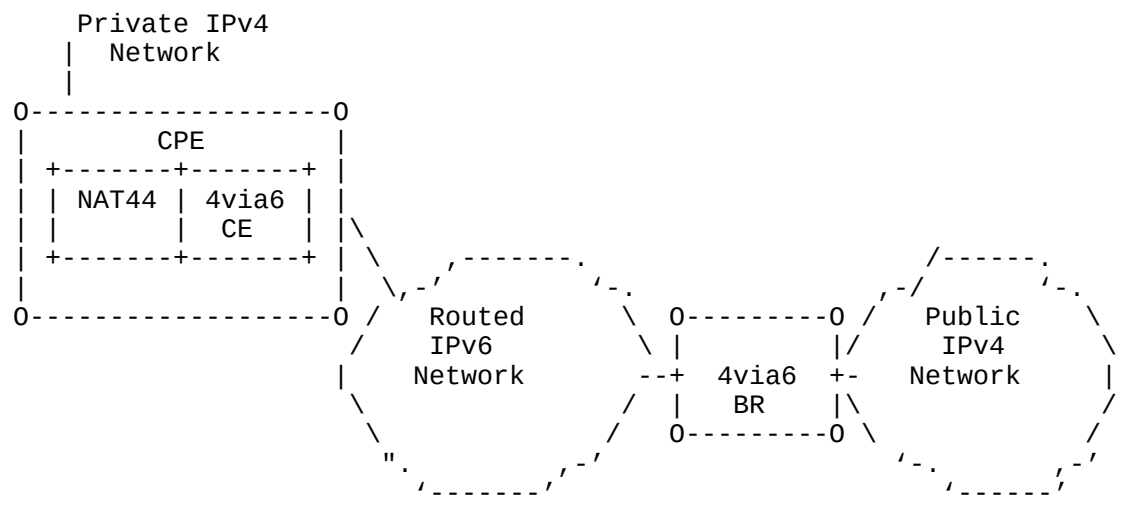


Figure 1: Network Topology

4via6 CE has two functionalities. The first is to generate an IPv4 address or an shared IPv4 address and port-set. The second is to translate an IPv4 packet from/to an IPv6 packet across IPv6 network.

When an unique IPv6 prefix is assigned to each CPE from SP's network, 4via6 CE in the CPE generates IPv4 address or shared IPv4 address and port-set with 4rd address mapping rule defined in [I-D.murakami-softwire-4rd].

The address mapping rule is also used in 4via6 CE to forward the

packets. When 4via6 CE sends a packet to BR, the source address is translated from IPv4 to IPv6 address with 4rd mapping rule and the destination address is translated from IPv4 to IPv6 address with [RFC6052]. In the case of sending the packet to another CE, the destination address is translated with 4rd address mapping rule.

NAT44 must be implemented in 4via6 CPE with the behavior conforming to the best current practice documented in [RFC4787], [RFC5508] and [RFC5382]. The NAT44 must translate the port number into the port-set generated in a given 4via6 CE.

At a BR side, when the BR sends a packet to a CE, the source address is translated from IPv4 to IPv6 address with [RFC6052] and the destination address is translated from IPv4 to IPv6 with 4rd mapping rule.

5. Stateless Translation Algorithm

The stateless translation between IPv6 and IPv4 must conform to [RFC6145]. The address mapping rule must be based on [I-D.murakami-softwire-4rd] and [RFC6052].

In 4via6 stateless translation, the only difference is the forwarding mechanism across IPv6 network infrastructure. The automatic tunneling mechanism such as IPv4-in-IPv6 is used in [I-D.murakami-softwire-4rd]. Instead, for the outband direction, the source address is translated with 4rd mapping rule and the destination address is translated with [RFC6052]. From the inbound direction, the source address is translated with [RFC6052] and the destination address is translated with 4rd mapping rule. For the direct communication among CEs, both source address and destination address are translated with only 4rd mapping rule.

6. Behavior of 4via6 Stateless Translation

6.1. Behavior on 4via6 CE

A 4via6 CE that receives IPv4 packets from CE LAN side checks the validity of its source and destination address. It also checks that the packet size is acceptable. If yes, NAT44 changes the IPv4 source address and the source port to its generated global IPv4 address and the port within the generated port-range. After that, 4via6 CE performs the translation of IPv4 source address and IPv4 destination address. The IPv4 source address is changed to the IPv6 address that is assigned to the 4via6 CE. The IPv4 destination address is translated based on [RFC6052]. And the IPv4 header is replaced to

the IPv6 header that is generated from the IPv4 header based on [RFC6145].

The 4via6 CE that receives IPv6 packet from CE WAN side checks the validity of its source and destination address. It also checks that the packet size is acceptable. If yes, it translates the IPv6 source and the IPv6 destination address in the received packets. The IPv6 destination address is changed to the IPv4 address that is generated in the 4via6 CE based on [I-D.murakami-softwire-4rd]. The IPv6 source address is translated based on [RFC6052]. After that, the IPv6 header is replaced to the IPv4 header that is generated from the IPv6 header based on [RFC6145].

6.2. Behavior on 4via6 BR

A 4rd BR that receives IPv4 packets from the outside IPv4 network checks the validity of its source and destination address. It also checks that the packet size is acceptable. If yes, it generates the IPv6 destination address from the IPv4 destination address based on [I-D.murakami-softwire-4rd] and translates the IPv4 source address to the IPv6 source address based on [RFC6052]. As the result, the IPv4 header is replaced to the IPv6 header based on [RFC6145].

The 4rd BR that receives IPv6 packets from IPv6 network infrastructure checks the validity of its source and destination address. It also checks that the packet size is accpetable. If yes, it generates the IPv4 source address from the IPv6 source address based on [I-D.murakami-softwire-4rd] and translates the IPv6 destination address to the IPv4 destination address based on [RFC6052]. As the result, the IPv6 header is replaced to the IPv4 header based on [RFC6145].

7. Path MTU and Fragmentation Consideration

Basically, Path MTU and Fragmentation must confirm to Section 1.4 of [RFC6145].

In 4via6 stateless transition, a 4via6 BR and a 4via6 CE replace an IPv6 header to an IPv4 header in a received IPv6 packet upon forwarding the packet to a native IPv4 interface. If the size of the IPv4 packet might exceed to the IPv4 MTU on the native IPv4 interface, the 4via6 BR and the 4via6 CE might fragment the packet. In order for the receiver to reassemble the fragmented packet correctly, the 4via6 BR and the 4via6 CE must assign an unique value to a datagram ID in IPv4 header upon forwarding the packet to the native IPv4 interface.

8. Comparison with 4rd

Differing from encapsulation model, translation approach doesn’t need to know BR IPv6 address. Instead of that, a IPv6 mapping prefix should be delivered to 4via6 CPEs or 4via6 hosts for generating IPv6 address by catenating IPv4 destination address with IPv6 mapping prefix. Such IPv6 mapping prefix shall be either the "Well-Known Prefix" or a "Network-Specific Prefix" unique to the organization deploying the address translators.

9. Security Considerations

The security consideration is same as [I-D.murakami-softwire-4rd].

10. IANA Consideration

This document has no IANA actions.

11. Acknowledgements

12. References

12.1. Normative References

[I-D.murakami-softwire-4rd]
Murakami, T. and T. Troan, "IPv4 Residual Deployment on IPv6 infrastructure - protocol specification (work in progress)", June 2011.

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

[RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.

12.2. Informative References

[I-D.despres-softwire-sam]
Despres, R., "Stateless Address Mapping (SAM) - a Simplified Mesh-Softwire Model", draft-despres-softwire-sam-01 (work in progress), July 2010.

[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-11 (work in progress), May 2011.

[I-D.operators-softwire-stateless-4v6-motivation]
Boucadair, M., Matsushima, S., Lee, Y., Bonness, O., Borges, I., and G. Chen, "Motivations for Stateless IPv4 over IPv6 Migration Solutions", draft-operators-softwire-stateless-4v6-motivation-02 (work in progress), June 2011.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC3513] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

[RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.

[RFC5508] Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT Behavioral Requirements for ICMP", BCP 148, RFC 5508, April 2009.

Authors' Addresses

Tetsuya Murakami (editor)
IP Infusion
1188 East Arques Avenue
Sunnyvale
USA

Email: tetsuya@ipinfusion.com

Gang Chen
China Mobile
53A,Xibianmennei Ave.,
Xuanwu District,
Beijing 100053
China

Email: chengang@chinamobile.com

Hui Deng
China Mobile
53A,Xibianmennei Ave.
Beijing 100053
P.R.China

Phone: +86-13910750201
Email: denghui02@gmail.com

Wojciech Dec
Cisco Systems
Haarlerbergpark Haarlerbergweg 13-19
Amsterdam, NOORD-HOLLAND 1101 CH
Netherlands

Phone:
Email: wdec@cisco.com

Satoru Matsushima
SoftBank Telecom
1-9-1 Higashi-Shinbashi, Munato-ku
Tokyo
Japan

Email: satoru.matsushima@tm.softbank.co.jp

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 11, 2011

M. Boucadair, Ed.
France Telecom
S. Matsushima
Softbank Telecom
Y. Lee
Comcast
O. Bonness
Deutsche Telekom
I. Borges
Portugal Telecom
G. Chen
China Mobile

June 9, 2011

Motivations for Stateless IPv4 over IPv6 Migration Solutions
draft-operators-softwire-stateless-4v6-motivation-02

Abstract

IPv4 service continuity is one of the most sensitive problems that must be resolved by Service Providers during the IPv6 transition period - especially after the exhaustion of the public IPv4 address space. Current standardization effort that addresses IPv4 service continuity focuses on stateful mechanisms. This document elaborates on the motivations for the need to undertake a companion effort to specify stateless IPv4 over IPv6 approaches.

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 11, 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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

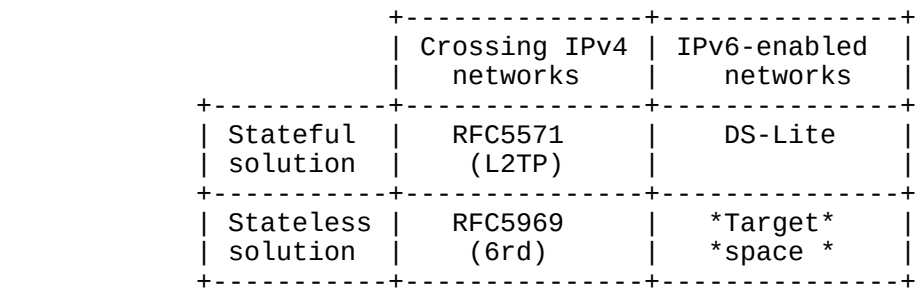
1. Introduction

When the global IPv4 address space is exhausted, Service Providers will be left with an address pool that cannot be increased anymore. Many services and network scenarios will be impacted by the lack of IPv4 public addresses. Providing access to the (still limited) IPv6 Internet only won't be sufficient to address the needs of customers, as most of them will continue to access legacy IPv4-only services. Service Providers must guarantee their customers that they can still access IPv4 contents although they will not be provisioned with a global IPv4 address anymore. Means to share IPv4 public addresses are unavoidable [I-D.ietf-intarea-shared-addressing-issues].

Identifying the most appropriate solution(s) to the IPv4 address exhaustion as well as IPv4 service continuity problems and deploying them in a real network with real customers is a very challenging and complex process for all Service Providers. There is nothing like a "One size fits all" solution or one target architecture that would work for all situations. Each Service Provider has to take into account its own context (e.g., service infrastructures), policies and marketing strategy (a document that informs Service Providers about the impact of the IPv4 address shortage, and provides some recommendations and guidelines, is available at [EURESCOM]).

Current standardization effort that is meant to address this IPv4 service continuity issue focuses mainly on stateful mechanisms that assume the sharing of any global IPv4 address that is left between several customers, based upon the deployment of NAT (Network Address Translation) capabilities in the network. Because of some caveats of such stateful approaches the Service Provider community feels that a companion effort is required to specify stateless IPv4 over IPv6 approaches. This document provides elaboration on such need.

Particularly, this document describes the motivations for stateless solutions within the context of an IPv6-enabled network as described in [RFC6180]. The following table shows the targeted space:



It is explicitly acknowledged by the authors of this document that both stateful and stateless solutions are required to meet Service Providers needs and constraints.

More discussions about stateless vs. stateful can be found at [RFC6144].

2. Terminology

This document makes use of the following terms:

Stateful 4/6 solution (or stateful solution in short): denotes a solution where the network maintains user-session states relying on the activation of a NAT function in the Service Providers' network [I-D.ietf-behave-lsn-requirements]. The NAT function is responsible for sharing the same IPv4 address among several subscribers and to maintain user-session state.

Stateless 4/6 solution (or stateless solution in short): denotes a solution which does not require any user-session state (see Section 2.3 of [RFC1958]) to be maintained by any IP address sharing function in the Service Provider's network. This category of solutions assumes a dependency between an IPv6 prefix and IPv4 address. In an IPv4 address sharing context, dedicated functions are required to be enabled in the CPE router to restrict the source IPv4 port numbers. Within this document, "port set" and "port range" terms are used interchangeably.

3. Why Stateless IPv4 over IPv6 Solutions are Needed?

This section discusses motivations for preferring a deployment of stateless 4/6 solutions. The technical and operational benefits of the stateless solutions are possible because no per-user state [RFC1958] is maintained in the Service Providers networks.

3.1. Network Architecture Simplification

The activation of this stateless function in the Service Provider's network does not introduce any major constraint on the network architecture and its engineering. The following sub-sections elaborate on these aspects.

3.1.1. Network Dimensioning

Because no user-state [RFC1958] is required, a stateless solution does not need to take into account the maximum number of simultaneous user-sessions and the maximum number of new user-sessions per second to dimension its networking equipment. Like current network dimensioning practices, only considerations related to the customer number, traffic trends and the bandwidth usage need be taken into account for dimensioning purposes.

3.1.2. No Intra-domain Constraint

Stateless IPv4/IPv6 interconnection functions can be ideally located at the boundaries of an Autonomous System (e.g., ASBR routers that peer with external IPv4 domains); in such case:

Intra-domain paths are not altered: there is no need to force IP packets to cross a given node for instance; intra-domain routing processes are not tweaked to direct the traffic to dedicated nodes. In particular, stateless solutions optimizes CPE-to-CPE communication in that packets don't go through the interconnection function since the address and port mapping has been realized based on a well defined mapping schema that is known to all involved devices.

3.1.3. Logging - No Need for Dynamic Binding Notifications

Network abuse reporting requires traceability [I-D.ietf-intarea-shared-addressing-issues]. To provide such traceability, prior to IPv4 address sharing, logging the IPv4 address assigned to a user was sufficient and generates relatively small logs. The advent of stateful IPv4 address allows dynamic port assignment, which then requires port assignment logging. This logging of port assignments can be considerable.

In contrast, static port assignments do not require such considerable logging. The volume of the logging file may not be seen as an important criterion for privileging a stateless approach because stateful approaches can also be configured (or designed) to assign port ranges and therefore lead to acceptable log volumes.

If a dynamic port assignment mode is used, dedicated interfaces and protocols must be supported to forward binding data records towards dedicated platforms. The activation of these dynamic notifications may impact the performance of the dedicated device. For stateless solutions, there is no need for dynamic procedures (e.g., using SYSLOG) to notify a mediation platform about assigned bindings.

Some Service Providers have a requirement to use only existing logging systems and to avoid introducing new ones (mainly because of CAPEX considerations). This requirement is easily met with stateless solutions.

3.1.4. No Additional Protocol for Port Control is Required

The deployment of stateless solution does not require the deployment of new dynamic signaling protocols to the end-user CPE in addition to those already used. In particular, existing protocols (e.g., UPnP IGD:2 [UPnP-IGD]) can be used to control the NAT mapping in the CPE.

3.1.5. Bandwidth Saving

In same particular network scenarios (e.g., wireless network), spectrum is very valuable and scarce resource. Service providers usually wish to eliminate unnecessary overhead to save bandwidth consumption in such environment. Service providers need to consider optimizing the form of packet processing when encapsulation is used. Since existing header compression techniques are stateful, it is expected that stateless solution minimize overhead introduced by the solution.

3.2. Operational Tasks and Network Maintenance Efficiency

3.2.1. Preserve Current Practices

Service Providers require as much as possible to preserve the same operations as for current IP networking environments.

If stateless solutions are deployed, common practices are preserved. In particular, the maintenance and operation of the network do not require any additional constraints such as: path optimization practices, enforcing traffic engineering policies, issues related to traffic oscillation between stateful devices, load-balancing the

traffic or load sharing the traffic among egress/ingress points can be used, etc. In particular:

- o anycast-based schemes can be used for load-balancing and redundancy purposes.

- o asymmetric routing to/from the IPv4 Internet is natively supported and no path-pinning mechanisms have to be additionally implemented.

3.2.2. Planned Maintenance Operations

Since no state is maintained by stateless IPv4/IPv6 interconnection nodes, no additional constraint needs to be taken into account when upgrading these nodes (e.g., adding a new service card, upgrading hardware, periodic reboot of the devices, etc.). In particular, current practices that are enforced to (gracefully) reboot or to shutdown routers can be maintained.

3.2.3. Reliability and Robustness

Compared to current practices (i.e., without a CGN in place), no additional capabilities are required to ensure reliability and robustness in the context of stateless solutions. Since no state is maintained in the Service Provider’s network, state synchronization procedures are not required.

High availability (including failure recovery) is ensured owing to best current practices in the field.

3.2.4. Support of Multi-Vendor Redundancy

Deploying stateful techniques, especially when used in the Service Providers networks, constrain severely deploying multi-vendor redundancy since very often proprietary vendor-specific protocols are used to synchronize state. This is not an issue for the stateless case. Concretely, the activation of the stateless IPv4/IPv6 interconnection function does not prevent nor complicate deploying devices from different vendors.

This criterion is very important for Service Providers having a sourcing policy to avoid mono-vendor deployments and to operate highly-available networks composed on multi-vendors equipment.

3.2.5. Simplification of Qualification Procedures

The introduction of new functions and nodes into operational networks follows strict procedures elaborated by Service Providers. These

procedures include in-lab testing and field trials. Because of their nature, stateless implementations optimize testing times and procedures:

- The specification of test suites to be conducted should be shorter;
- The required testing resources (in terms of manpower) are likely to be less solicited that they are for stateful approaches.

One of the privileged approaches to integrate stateless IPv4/IPv6 interconnection function consists in embedding stateless capabilities in existing operational nodes (e.g., IP router). In this case, any software or hardware update would require to execute non-regression testing activities. In the context of the stateless solutions, the non-regression testing load due to an update of the stateless code is expected to be minimal.

For the stateless case, testing effort and non-regression testing are to be taken into account for the CPE side. This effort is likely to be lightweight compared to the testing effort, including the non-regression testing, of a stateful function which is co-located with other routing functions for instance.

3.3. Facilitating Service Evolution

3.3.1. Implicit Host Identification

Service Providers do not offer only IP connectivity services but also added value services (a.k.a., internal services). Upgrading these services to be IPv6-enabled is not sufficient because of legacy devices. In some deployments, the delivery of these added-value services relies on implicit identification mechanism based on the source IPv4 address. Due to address sharing, implicit identification will fail [I-D.ietf-intarea-shared-addressing-issues]; replacing implicit identification with explicit authentication will be seen as a non acceptable service regression by the end users (less Quality of Experience (QoE)).

When a stateless solution is deployed, implicit identification for internal services is likely to be easier to implement: the implicit identification should be updated to take into account the port range and the IPv4 address. Techniques as those analyzed in [I-D.boucadair-intarea-nat-reveal-analysis] are not required for the delivery of these internal services if a stateless solution is deployed.

3.3.2. No Organizational Impact

Stateless solutions rely on IP-related techniques to share and to deliver IPv4 packets over an IPv6 network. In particular, IPv4 packets are delivered without any modification to their destination CPE. As such there is a clear separation between the IP/transport layers and the service layers; no service interference is to be observed when a stateless solution is deployed. This clear separation:

Facilitates service evolution: Since the payload of IPv4 packets is not altered in the path, services can evolve without requiring any specific function in the Service Provider's network;

Limits vendor dependency: The upgrade of value-added services does not involve any particular action from vendors that provide devices embedding the stateless IPv4/IPv6 interconnection function.

No service-related skills are required for network operators who manage devices that embed the IPv4/IPv6 interconnection function: IP teams can be in charge of these devices; there is a priori no need to create a dedicated team to manage and to operate devices embedding the stateless IPv4/IPv6 interconnection function. The introduction of stateless capabilities in the network are unlikely to degrade management costs.

3.4. Cost Minimization Opportunities

To make decision for which solution is to be adopted, service providers usually undertake comparative studies about viable technical solutions. It is not only about technical aspects but also economical optimization (both CAPEX and OPEX considerations).

From a Service Provider perspective, stateless solutions are more attractive because they do less impact the current network operations and maintenance model that is widely based on stateless approaches. Table 1 shows the general correspondence between technical benefits and potential economic reduction opportunities.

While not all Service Providers environments are the same, a detailed case study from one Service Provider [I-D.matsushima-v6ops-transition-experience] reports that stateless transition solutions can be considerably less expensive than stateful transition solutions.

Section	Technical and Operation Benefit	Cost Area
Section 3.1.1	Network dimensioning	Network
Section 3.1.2	No Intra-domain constraint	Network
Section 3.1.3	Logging	Network & Ops
Section 3.1.4	No additional control protocol	Network
Section 3.2.1	Preserve current practices	Ops
Section 3.2.2	Planned maintenance	Ops
Section 3.2.3	Reliability and robustness	Network & Ops
Section 3.2.4	Multi-Vendor Redundancy	Network
Section 3.2.5	Simple qualification	Ops
Section 3.3.1	Implicit Host Identification for internal services	Ops
Section 3.3.2	Organizational Impact	Ops

Table 1: Cost minimization considerations

4. Conclusion

As discussed in Section 3, stateless solutions provide several interesting features. Trade-off between the positive vs. negative aspects of stateless solutions is left to Service Providers. Each Service Provider will have to select the appropriate solution (stateless, stateful or even both) meeting its requirements.

This document recommends to undertake as soon as possible the appropriate standardization effort to specify a stateless IPv4 over IPv6 solution.

5. IANA Considerations

No action is required from IANA.

6. Security Considerations

Except for the less efficient port randomization of and routing loops [I-D.ietf-v6ops-tunnel-loops], stateless 4/6 solutions are expected to introduce no more security vulnerabilities than stateful ones. Because of their stateless nature, they may in addition reduce denial of service opportunities.

7. Contributors

The following individuals have contributed to this document:

Christian Jacquenet
France Telecom

Email: christian.jacquenet@orange-ftgroup.com

Pierre Levis
France Telecom

Email: pierre.levis@orange-ftgroup.com

Masato Yamanishi
SoftBank BB

Email: myamanis@bb.softbank.co.jp

Yuji Yamazaki
Softbank Mobile

Email: yuyamaza@bb.softbank.co.jp

Hui Deng
China Mobile
53A,Xibianmennei Ave.
Beijing 100053
P.R.China

Phone: +86-13910750201
Email: denghui02@gmail.com

8. Acknowledgments

Many thanks to the following individuals who provided valuable comments:

X. Deng	W. Dec	D. Wing	A. Baudot
E. Burgey	L. Cittadini	R. Despres	J. Zorz
M. Townsley	L. Meillarec	R. Maglione	J. Queiroz
C. Xie	X. Li	O. Troan	J. Qin
B. Sarikaya			

9. Informative References

[EURESCOM]
Levis, P., Borges, I., Bonness, O. and L. Dillon L., "IPv4 address exhaustion: Issues and Solutions for Service Providers", March 2010, <http://archive.eurescom.eu/~pub/deliverables/documents/P1900-series/P1952/D2bis/P1952-D2bis.pdf>.

[I-D.boucadair-intarea-nat-reveal-analysis]
Boucadair, M., Touch, J., and P. Levis, "Analysis of Solution Candidates to Reveal the Origin IP Address in Shared Address Deployments", draft-boucadair-intarea-nat-reveal-analysis-01 (work in progress), March 2011.

[I-D.ietf-behave-lsn-requirements]
Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for IP address sharing schemes", draft-ietf-behave-lsn-requirements-01 (work in progress), March 2011.

[I-D.ietf-intarea-shared-addressing-issues]
Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", draft-ietf-intarea-shared-addressing-issues-05 (work in progress), March 2011.

[I-D.ietf-v6ops-tunnel-loops]
Nakibly, G. and F. Templin, "Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", draft-ietf-v6ops-tunnel-loops-07 (work in progress), May 2011.

[I-D.matsushima-v6ops-transition-experience]
Matsushima, S., Yamazaki, Y., Sun, C., Yamanishi, M., and J. Jiao, "Use case and consideration experiences of IPv4 to IPv6 transition",

draft-matsushima-v6ops-transition-experience-02 (work in progress), March 2011.

[RFC1958] Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 1996.

[RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.

[RFC6180] Arkko, J. and F. Baker, "Guidelines for Using IPv6 Transition Mechanisms during IPv6 Deployment", RFC 6180, May 2011.

[UPnP-IGD] UPnP Forum, "Universal Plug and Play (UPnP) Internet Gateway Device (IGD) V 2.0", December 2010, <http://upnp.org/specs/gw/igd2/>.

Authors' Addresses

Mohamed Boucadair (editor)
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

Satoru Matsushima
Softbank Telecom
Tokyo
Japan

Email: satoru.matsushima@tm.softbank.co.jp

Yiu Lee
Comcast
US

Email: Yiu_Lee@Cable.Comcast.com

Olaf Bonness
Deutsche Telekom
Germany

Email: Olaf.Bonness@telekom.de

Isabel Borges
Portugal Telecom
Portugal

Email: Isabel@ptinovacao.pt

Gang Chen
China Mobile
53A,Xibianmennei Ave.
Beijing, Xuanwu District 100053
China

Email: chengang@chinamobile.com

Softwire WG
Internet-Draft
Intended status: Standards Track
Expires: December 15, 2011

Q. Wang
China Telecom
J. Qin
ZTE
M. Boucadair
C. Jacquenet
France Telecom
Y. Lee
Comcast
June 13, 2011

Multicast Extensions to DS-Lite Technique in Broadband Deployments
draft-qin-softwire-dslite-multicast-04

Abstract

This document proposes a solution for the delivery of multicast service offerings to DS-Lite serviced customers. The proposed solution relies upon a stateless IPv4-in-IPv6 encapsulation scheme and does not require performing any NAT operation along the path used to deliver multicast traffic.

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 15, 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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

DS-Lite [I-D.ietf-softwire-dual-stack-lite] is a technique to rationalize the use of the remaining IPv4 addresses during the transition period. The current design of DS-Lite covers unicast services exclusively.

If customers access IPv4 multicast-based service offerings through a DS-Lite environment, AFTR (Address Family Transition Router) devices have to process all the IGMP reports [RFC2236] [RFC3376] received within IPv4-in-IPv6 tunnels and behave as a replication point for downstream multicast traffic. That is likely to severely affect the multicast traffic forwarding efficiency by losing the benefits of deterministic replication of the data as close to the receivers as possible. As a consequence, the downstream bandwidth will be vastly consumed while the AFTR capability may become rapidly overloaded, in particular if the AFTR capability is deployed in a centralized manner.

This document discusses an extension to the DS-Lite model to be used for the delivery of IPv4 multicast-based service offerings.

1.1. 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 RFC 2119 [RFC2119].

2. Terminology

This document makes use of the following terms:

- IPv4-embedded IPv6 address: is an IPv6 address which embeds a 32 bit-encoded IPv4 address. An IPv4-embedded IPv6 address can be unicast or multicast.

- mPrefix64: is a dedicated multicast IPv6 prefix for constructing IPv4-embedded IPv6 multicast address [I-D.boucadair-behave-64-multicast-address-format]. mPrefix64 can be of two types: ASM_mPrefix64 used in ASM mode or SSM_mPrefix64 used in SSM mode [RFC4607].

- uPrefix64: is a dedicated unicast IPv6 prefix for constructing IPv4-embedded IPv6 unicast address [RFC6052].

- Multicast AFTR (mAFTR for short): is a functional entity which is part of both the IPv4 and IPv6 multicast distribution trees and

which replicates IPv4 multicast streams into IPv4-in-IPv6 streams in the relevant branches of the IPv6 multicast distribution tree.

- Multicast B4 (mB4 for short): is a functional entity embedded in a CPE, which is able to enforce an IGMP-MLD interworking function (refer to Section 6.1) together with a de-capsulation function of received multicast IPv4-in-IPv6 packets.

3. Context and Scope

3.1. IPTV-centric View

IPTV generally includes two categories of service offerings:

1. VoD (Video on Demand) or Catch-up TV channels streams that are delivered using unicast mode to receivers.

2. Live TV Broadcast services that are generally multicast to receivers.

Numerous players intervene in the delivery of this service:

- Content Providers: the content can be provided by the same provider as the one providing the connectivity service or by distinct providers;

- Network Provider: the one providing network connectivity service (e.g., responsible for carrying multicast flows from head-ends to receivers). Refer to [I-D.ietf-mboned-multiaaa-framework].

Many of the current IPTV contents are likely to remain IPv4-formatted and out of control of the network providers. Additionally, there are numerous legacy receivers (e.g., IPv4-only Set Top Boxes (STB)) that can’t be upgraded or be easily replaced. As a consequence, IPv4 service continuity must be guaranteed during the transition period, including the delivery of multicast-based services such as Live TV Broadcasting. The dilemma is the same as in the transition of unicast-based Internet services where the customer premises and global Internet are out of control of the service providers even if they would like to promote the use of IPv6. The DS-Lite design tries to eliminate this issue by decoupling the IPv6 deployments in service provider networks from that in global Internet and in customer devices and applications.

DS-Lite can be seen as a catalyst for IPv6 deployment while preserving customer’s Quality of Experience (QoE). This is also the design goal of the solution proposed in this document for DS-Lite

serviced customers who have subscribed to a multicast-based service offering.

3.2. Scope

This document focuses only on issues raised by a DS-Lite networking environment: subscription to an IPv4 multicast group and the delivery of IPv4-formatted content to IPv4 receivers. In particular, only the following case is covered:

1. An IPv4 receiver accessing IPv4 content (i.e., content formatted and reachable in IPv4)

A viable scenario for this use case in DS-Lite environment: Customers with legacy receivers must continue to access the IPv4-enabled multicast services. This means the traffic should be accessed through IPv4 and additional functions are needed to traverse operators' IPv6- enabled network which is the purpose of this document. While since technically, there is no extra function required for the scenario of native access (i.e. to access dual-stack content natively from the IPv6 receiver), this portion is not taken into account. Refer to [I-D.jaclee-behave-v4v6-mcast-ps] for the deployment considerations.

This document does not cover the case where an IPv4 host connected to a CPE served by a DS-Lite AFTR can be the source of multicast traffic.

Note that some contract agreements prevent a network provider to alter the content as sent by the content provider, in particular for copyright, confidentiality and SLA assurance reasons. The streams should be delivered unaltered to requesting users.

4. Solution Overview

In the original DS-Lite specification [I-D.ietf-softwire-dual-stack-lite], an IPv4-in-IPv6 tunnel is used to carry the bidirectional IPv4 unicast traffic between B4 and AFTR. This document defines an IPv4-in-IPv6 encapsulation scheme to deliver multicast traffic. Within the context of this document, an IPv4 derived IPv6 multicast address is used as the destination of the encapsulated unidirectional IPv4-in-IPv6 multicast traffic from the mAFTR to the mB4. The IPv4 address of the source of the multicast content is represented in the IPv6 realm with an IPv4-embedded IPv6 address as well.

See following sections for the multicast distribution tree

establishment (Section 4.3) and the multicast traffic forwarding (Section 4.4).

Note that IPv4-in-IPv6 encapsulated multicast flows are treated in an IPv6 realm like any other IPv6 multicast flow. Upon completion of the establishment of a multicast distribution tree, no extra function is required to be defined to forward IPv4-in-IPv6 multicast traffic in the IPv6 network.

4.1. Rationale

This document introduces two new functional elements (Figure 1):

1. The mAFTR: responsible for replicating IPv4 multicast flows in the IPv6 domain owing to a stateless IPv4-in-IPv6 encapsulation function. The mAFTR does not undertake any NAT operation. The mAFTR is a demarcation point which connects to both the IPv4 and IPv6 multicast networks.

2. The mB4: is a functional entity embedded in a CPE responsible for the de-capsulation of the received IPv4-in-IPv6 multicast packets and forwarding them to the appropriate IPv4 receivers.

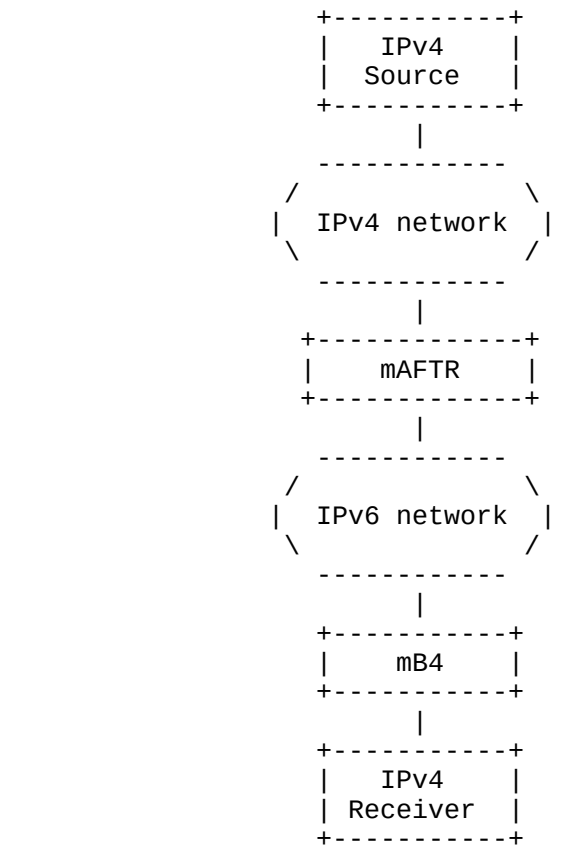


Figure 1: Functional Architecture

4.2. IPv4-embedded IPv6 Address Prefixes

A dedicated IPv6 multicast prefix (mPrefix64) is needed for forming IPv6 multicast addresses, with IPv4 multicast address embedded. The mPrefix64 can be of two types: ASM_mPrefix64 (an mPrefix64 used in ASM mode) or SSM_mPrefix64 (an mPrefix64 used in SSM mode), and MUST be derived from the corresponding IPv6 multicast address space [I-D.boucadair-behave-64-multicast-address-format].

In addition, the address of the IPv4 multicast source should be mapped to IPv6 addresses in the IPv6 realm: an IPv6 unicast prefix (uPrefix64) is therefore needed for forming IPv6 unicast addresses with IPv4 unicast address embedded. The uPrefix64 MUST be derived from the IPv6 unicast address space [RFC6052].

The mAFTR and mB4 MUST use the same mPrefixe64 and uPrefix64, and the

same algorithm for building IPv4-embedded IPv6 addresses. Refer to Section 5 for more details on the IPv6 address format.

4.3. Multicast Distribution Tree

Assume that an IPv4 receiver sends an IGMP Report towards the mB4 to join a given multicast group. After receiving the IGMP Report message, the mB4 converts the IGMP message into a MLD Report [RFC2710] message which will then be forwarded upstream towards the MLD Querier. The MLD Querier is likely to coexist with the PIM DR where the PIMv6 Join message will be triggered and sent up hop by hop along the PIMv6 routers. Note that the mAFTR is in the path to reach the IPv4 source; this is typically achieved by the underlying unicast IPv6 routing protocol that advertises the unicast IPv4-embedded IPv6 addresses: these addresses are used to represent IPv4 sources in the IPv6 multicast domain.

Both the MLD and the PIMv6 Join messages convey the IPv6 address of the multicast group to be joined. The corresponding IPv6 multicast group address is constructed by using the pre-configured mPrefix64 and an algorithm so that the IPv4 multicast group address is embedded accordingly.

When source-specific multicast is deployed, the IPv6 address of the multicast source should be constructed in the same way (using uPrefix64, with IPv4 multicast source embedded). Refer to Section 6.1 for more details of the mB4 function.

- If the mAFTR is embedded in the MLD Querier/PIMv6 DR, it should process the received MLD Report message for the IPv4-embedded IPv6 group and send the corresponding IPv4 PIM Join message.

- If the mAFTR is embedded in some upstream PIMv6 router more than one hop away from the mB4, it should process the received PIMv6 Join message for the IPv4-embedded IPv6 group and send the corresponding IPv4 PIM Join message.

In both cases, an entry for an IPv6 multicast group address is created by the mAFTR in its multicast Routing Information Base and is used to forward multicast IPv4-in-IPv6 datagrams. Refer to Section 7.1 for more details about the mAFTR function.

A branch of the multicast distribution tree is then established, comprising both an IPv4 part (from the mAFTR upstream) and an IPv6 part (between the mB4 and the mAFTR).

4.4. Multicast Forwarding

Whenever an IPv4 multicast packet is received on a mAFTR (this assumes the RPF Check has passed Section 7.1), it will be encapsulated into an IPv6 packet using the IPv4-embedded IPv6 multicast address as the destination address and an IPv4-embedded IPv6 unicast address as the source of the IPv4-in-IPv6 packet. The new IPv6 multicast packet will then be sent through the outgoing interface of the matching entry in the multicast routing table and forwarded down the IPv6 multicast distribution tree towards the mB4.

When receiving the packet, the mB4 should de-capsulate it and forward the original IPv4 multicast packet to the appropriate receiver. If mB4 does not have any route to forward the packet (e.g., change of the IPv4 address without cleaning MLD states), the encapsulated IPv4 datagram is silently dropped.

Note that: There is an alternative to the encapsulation based mechanism (which is detailed in this memo) for Multicast Forwarding: Translation based approach, which is per [I-D.boucadair-behave-64-multicast-address-format], [RFC6052] and [RFC6145]. Refer to Appendix A.

4.5. Multicast DS-Lite vs. Unicast DS-Lite

Unlike a unicast AFTR, a mAFTR does not perform any NAT for delivering IPv4 multicast traffic.

Unlike unicast DS-Lite, a mB4 does not need to discover a mAFTR.

mAFTR is responsible for encapsulating in a stateless manner the IPv4 multicast traffic into IPv6 datagrams. mB4 is responsible for de-capsulating in a stateless manner the IPv4-in-IPv6 multicast traffic. Further elaboration is provided in the following sections about the behaviour of the mAFTR and the mB4.

The corresponding multicast DS-Lite and the unicast DS-Lite functional elements can be co-located in the same device or separated.

5. Address Mapping

5.1. Prefix Assignment

In order to map the addresses of IPv4 multicast traffic with IPv6 multicast addresses, an IPv6 multicast prefix (mPrefix64) and an IPv6 unicast prefix (uPrefix64) are provided to mAFTR and mB4 elements.

The address format to be used is being left to the responsibility of the service provider as indicated in [RFC6052] and [I-D.boucadair-behave-64-multicast-address-format].

The mPrefix64 and uPrefix64 together with the address format to be used can be configured in the mB4 through a dedicated provisioning protocol, such as DHCPv6 or another protocol. Two candidate DHCPv6 options are identified in [I-D.ietf-behave-nat64-learn-analysis].

5.2. Text Representation Examples

Group address mapping example when a /96 is used:

mPrefix64	IPv4 address	IPv4-Embedded IPv6 address
ffxx:abc::/96	230.1.2.3	ffxx:abc::230.1.2.3

Source address mapping example when a /96 is used:

uPrefix64	IPv4 address	IPv4-Embedded IPv6 address
2001:db8::/96	192.1.2.3	2001:db8::192.1.2.3

6. Multicast B4 (mB4)

6.1. IGMP-MLD Interworking function

IGMP-MLD Interworking function combines the IGMP/MLD Proxying function specified in [RFC4605] and the IGMP/MLD adaptation function which is meant to reflect the contents of IGMP messages into MLD messages.

Then mB4 performs the router portion of the IGMP protocol on each downstream interface and performs the host portion of the MLD protocol on the upstream interface (Figure 2).

The output of the operation is a set of membership information which is maintained separately on each downstream interface (e.g., Wifi and Wired Ethernet). In addition, the membership information on each downstream interface is merged into the membership database on which the IPv4 multicast packets are forwarded by mB4.

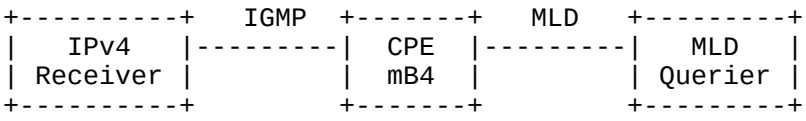


Figure 2: IGMP-MLD Interworking

When an IGMP Report message is received from a receiver to subscribe to a given multicast group G (e.g., 230.1.2.3) (and optionally associated to a source 192.1.2.3 if SSM mode is used), the mB4 MUST send an MLD Report message to subscribe to the corresponding IPv6 group identified by an IPv4-embedded IPv6 multicast address using a pre-configured prefix and algorithm (e.g., ffxx:abc::230.1.2.3 (and optionally source 2001:db8::192.1.2.3 if SSM mode is used)). The MLD Report message is sent through the upstream interface natively (i.e., without any encapsulation).

6.2. De-capsulation and Forwarding

When the mB4 receives an IPv6 multicast packet, it checks whether the group address is in the range of mPrefix64 and the source address is in the range of uPrefix64. If it is true, the mB4 MUST de-capsulate the IPv4-in-IPv6 packets to extract the original IPv4 multicast packets.

Then the IPv4 multicast packet will be forwarded to downstream receivers based on information maintained by the mB4 in the membership database. If no route is found, the packet is silently dropped.

6.3. Fragmentation

Encapsulating IPv4 over IPv6 from mAFTR to mB4 for data forwarding reduces the effective MTU size by the size of an IPv6 header (assuming [RFC2473] encapsulation). To avoid fragmentation, a service provider may increase the MTU size by 40 bytes on the IPv6 network or mAFTR and mB4 may use IPv6 Path MTU discovery.

6.4. Host with mB4 function embedded

The mB4 function can be embedded in the CE or in a dual-stack host behind the CP router (e.g., STB). If mB4 is embedded in the STB, the IGMP-MLD interworking function is not needed. The STB should formulate the MLD message correspondingly based on given IPv4 group address to be joint using mPrefix64 (and uPrefix64 for IPv4-embedded source if SSM is deployed), and de-encapsulate the downstream multicast traffics received by itself.

7. Multicast AFTR (mAFTR)

7.1. Routing Considerations

Except the need for the mAFTR to belong to IPv4 multicast distribution trees and to be on the reverse path towards the source when performing RPF checks on PIMv6 routers, no further routing constraint is to be taken into account.

Having the mAFTR in the reverse path ensures PIM Join sent to the source (e.g., SSM mode or SPT mode in ASM) will be intercepted by the mAFTR.

7.2. Processing PIM/MLD Join Messages

Upon receipt of the PIM/MLD Join for an IPv6 group (e.g., ffxx:abc::230.1.2.3), the mAFTR checks the corresponding entry in the IPv6 multicast routing table and adds the IPv6 interface through which the Join message has been received into the Out-Interface-List of that entry. If the entry does not exist, a new one will be created, as per typical PIM machinery [RFC4601]. The mAFTR should check whether the IPv6 group address belongs to the mPrefix64 (e.g., ffxx:abc::/96). If so, the mAFTR will need to extract the IPv4 group address (e.g., 230.1.2.3) from the IPv4-embedded IPv6 address (e.g., according to [I-D.boucadair-behave-64-multicast-address-format]) and check the corresponding entry in the IPv4 multicast routing table then add the tunnel interface into the Out-Interface-List of that entry. If the entry does not exist, a new entry should be created and a PIM join message for that IPv4 group will be sent towards the RP or source connected to the IPv4 network.

When SSM is deployed, the mAFTR would in addition check if the source (e.g., 2001:db8::192.1.2.3) described in the PIMv6 Join message belongs to uPrefix64 (e.g., 2001:db8::/96). If so, it can then send a PIM (S, G) Join message directly towards the IPv4 source (e.g., 192.1.2.3).

The initialization of the tunnel interface (used for encapsulation purposes) on the mAFTR is out of the scope of this document.

7.3. Reliability

For robustness, reliability and load distribution purposes, several nodes in the network can embed the mAFTR function. In such case, the same IPv6 prefixes (i.e., mPrefix64 and uPrefix64) and algorithm to build IPv4-embedded IPv6 addresses MUST be configured on those nodes.

7.4. ASM Mode: Building Shared Trees

7.4.1. IPv4 Side

For a given Rendezvous Point (RP) used in the IPv4 realm, there is no new requirement. Like any other IPv4 PIM router, the RP of each IPv4 multicast groups is configured to the mAFTR or discovered using some appropriate means. Moreover, PIM-SM registration procedure [RFC4601] in the IPv4 realm is not impacted.

Shared IPv4 multicast trees are built using the procedure defined in [RFC4601] for instance.

7.4.2. IPv6 Side

In the IPv6 side, the RP of IPv4-embedded IPv6 multicast groups is configured to all IPv6 PIM routers or discovered using appropriate means. For the sake of simplicity, it is RECOMMENDED to configure an mAFTR as the RP for IPv4-embedded IPv6 multicast groups.

[Note 1: If some other IPv6 multicast router wants to become the RP of the IPv4-embedded IPv6 multicast groups, it may require an mAFTR to emulate the PIM Source Register procedure on behalf of IPv4-embedded IPv6 sources with the RP. The PIM Source Register procedure in the IPv4 domain is not altered.]

[Note 2: How the mAFTR is aware about the sources? This can be considered as deployment-specific:

(i) By configuration: mAFTR can be configured to join a set of IPv4 multicast groups and to initiate a registration procedure on behalf of a set of sources to the RP in the v6 domain;

(ii) Dynamic: this assumes that mAFTR is configured to join a set of IPv4 multicast groups. The source address of received flows will be used as a trigger to initiate the registration procedure to the RP in the IPv6 domain. There is a special case where mAFTR is the RP of the IPv4 group in the IPv4 domain: The registration procedure should then be relayed to the RP in the IPv6 domain.

]

Shared IPv6 multicast trees are built using the procedure defined in [RFC4601] for instance. Switching from a shared tree to source-based tree can be accommodated since the mAFTR is in the path to join the source.

The mAFTR will graft to the IPv4 shared tree either because it has been configured with the list of IPv4 multicast groups that will be subscribed by the DS-Lite serviced receivers downstream or upon receipt of a PIMv6 Join message.

An example of the exchange of PIM messages is illustrated in Figure 3.

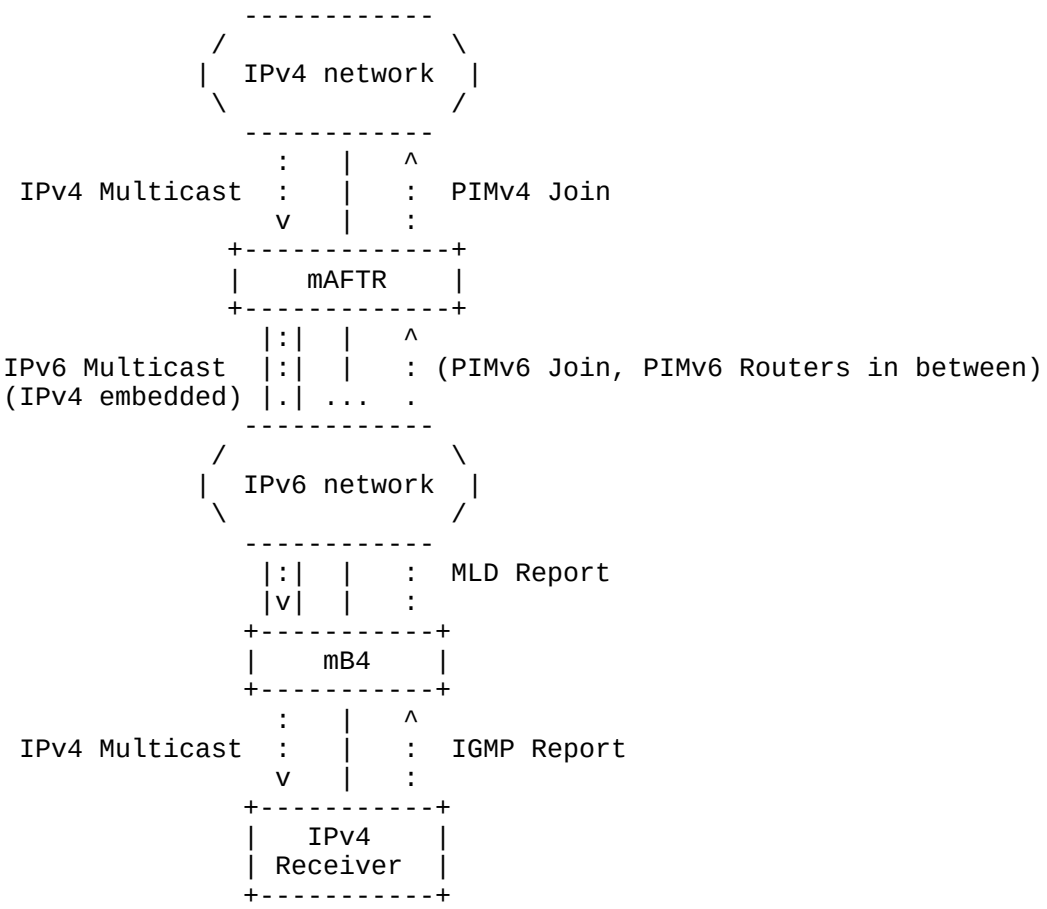


Figure 3: Procedure Overview

7.5. TTL/Scope

The Scope field of IPv4-in-IPv6 multicast addresses can be valued to "E" (Global scope) or to "8" (Organization-local scope). This is left to service providers taste.

7.6. Encapsulation and forwarding

When receiving an IPv4 multicast packet, a lookup of the IPv4 multicast routing table is performed by the PIMv4 router that embeds the mAFTR capability. If an interface used for IPv4-in-IPv6 encapsulation is found in the Out-Interface-List of the matching entry, the encapsulation operation is triggered. The mAFTR encapsulates in a stateless fashion the IPv4 multicast packet into an IPv6 multicast datagram. It MUST use the pre-provisioned mPrefix64/ uPrefix64 together with an algorithm for building the IPv4-embedded IPv6 multicast address that identifies the multicast group, as well as the IPv6 source address that represents the IPv4 source in the IPv6 network.

As an illustration, if a packet is received from source 192.1.2.3 and forwarded to group 230.1.2.3, the mAFTR encapsulates it into an IPv6 multicast packet using ffxx:abc::230.1.2.3 as the destination IPv6 address and 2001:db8::192.1.2.3 as the multicast source address.

Then a lookup of the IPv6 multicast routing table is performed by the PIMv6 router that embeds the mAFTR capability, based on the IPv4-embedded IPv6 address. If a matching entry is found and there exist IPv6 interfaces in the Out-Interface-List, the IPv6 multicast packet will be sent out through these interfaces and forwarded down the multicast distribution tree towards the mB4 devices.

8. Optimization in L2 Access Networks

The approach specified in this document is compatible with a Layer-2 infrastructure which may be involved for deterministic multicast replication.

The IPv4-in-IPv6 encapsulated multicast flows destined to IPv4-embedded IPv6 group addresses are treated as any IPv6 multicast flow, and can be replicated across Multicast VLANs. Additionally, mechanisms such as MLD Snooping, MLD Proxying, etc., can be introduced into the distributed Access Network Nodes (e.g., Aggregation Switches, xPON devices) which then could behave as MLD Querier and replicate multicast flows as appropriate. Thus, the multicast replication point is moved downward closer to the receivers, so that bandwidth consumption is optimized.

9. Security Considerations

This document does not introduce any new security concern in addition to what is discussed in Section 5 of [RFC6052], Section 10 of

[RFC3810] and Section 6 of [RFC4601].

9.1. Firewall Configuration

The CPE should be configured to accept incoming MLD messages and traffic forwarded to multicast groups subscribed by receivers located in the customer premises.

10. Acknowledgements

The authors would like to thank Dan Wing for his guidance in the early discussions which initiated this work. We also appreciate Peng Sun, Jie Hu, Qiong Sun, Lizhong Jin, Alain Durand, Dean Cheng, and Behcet Sarikaya for their valuable comments.

11. IANA Considerations

This document includes no request to IANA.

12. References

12.1. Normative References

[I-D.boucadair-behave-64-multicast-address-format]
Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X., and M. Xu, "IPv4-Embedded IPv6 Multicast Address Format", draft-boucadair-behave-64-multicast-address-format-01 (work in progress), February 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-11 (work in progress), May 2011.

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

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

[RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

[RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.

12.2. Informative References

[I-D.ietf-behave-nat64-learn-analysis]
Korhonen, J. and T. Savolainen, "Analysis of solution proposals for hosts to learn NAT64 prefix", draft-ietf-behave-nat64-learn-analysis-00 (work in progress), May 2011.

[I-D.ietf-mboned-multiaaa-framework]
Satou, H., Ohta, H., Hayashi, T., Jacquenet, C., and H. He, "AAA and Admission Control Framework for Multicasting", draft-ietf-mboned-multiaaa-framework-12 (work in progress), August 2010.

[I-D.jaclee-behave-v4v6-mcast-ps]
Jacquenet, C., Boucadair, M., Lee, Y., Qin, J., and T. ZOU), "IPv4-IPv6 Multicast: Problem Statement and Use Cases", draft-jaclee-behave-v4v6-mcast-ps-02 (work in progress), June 2011.

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.

[RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, August 2006.

[RFC4608] Meyer, D., Rockell, R., and G. Shepherd, "Source-Specific Protocol Independent Multicast in 232/8", BCP 120, RFC 4608, August 2006.

Appendix A. Translation vs. Encapsulation

In order to deliver IPv4 multicast flows to DS-Lite serviced receivers, two options can be considered:(1) Translation; (2)Encapsulation.

It should be noted that some contract agreement may prevent the contents from being altered. In this case, the employment of the translation approach may raise issues e.g., Integrity Check failures.

A.1. Translation

To delivery IPv4 multicasst contents to an IPv4 receiver: Introduce translation functions at the boundaries of IPv6 network. The IPv4-translated multicast streams are distributed within the IPv6 network natively until the customer premises device where the IPv4-translated IPv6 streams are translated back and passed to IPv4 receivers. Multicast Distribution Tree is established by normal machinery of control protocols (e.g. IGMP, MLD, PIMv4/v6) and the Interworking functions (e.g. IGMP-MLD, PIMv6-PIMv4), refer to Section 6 and Section 7. The translation function is stateless owing to the use of IPv4-Embedded IPv6 address [I-D.boucadair-behave-64-multicast-address-format] and [RFC6052].

A.2. Encapsulation

To deliver IPv4 multicast contents to an IPv4 receiver: Introduce two elements at the boundaries of IPv6 network, mAFTR and mB4. Multicast Distribution Tree is established by normal machinery of control protocols (e.g. IGMP, MLD, PIMv4/v6) and the Interworking functions (e.g. IGMP-MLD, PIMv6-PIMv4), refer to Section 6 and Section 7. Multicast streams are forwarded to a receiver by using an IPv4-in-IPv6 encapsulation scheme. The encapsulation/de-capsulation function is stateless owing to the use of IPv4-Embedded IPv6 address [I-D.boucadair-behave-64-multicast-address-format] and [RFC6052].

Authors' Addresses

Qian Wang
China Telecom
No.118, Xizhimennei
Beijing, 100035
China

Phone: +86 10 5855 2177
Email: wangqian@ctbri.com.cn

Jacni Qin
ZTE
Shanghai,
China

Phone: +86 1391 8619 913
Email: jacniq@gmail.com

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Phone:
Email: mohamed.boucadair@orange-ftgroup.com

Christian Jacquenet
France Telecom
Rennes, 35000
France

Phone:
Email: christian.jacquenet@orange-ftgroup.com

Yiu L. Lee
Comcast
U.S.A.

Phone:
Email: yiu_lee@cable.comcast.com
URI: http://www.comcast.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 12, 2012

S. Tsuchiya, Ed.
M. Townsley
Cisco Systems
S. Ohkubo
Sakura Internet
July 11, 2011

IPv6 Rapid Deployment (6rd) in a Large Data Center
draft-sakura-6rd-datacenter-01

Abstract

IPv6 Rapid Deployment (6rd) as defined in RFC 5969 focuses on rapid deployment of IPv6 by an access service provider which has difficulty deploying native IPv6. This document describes how 6rd can be used to deliver IPv6 within a Large Data Center.

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

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

Table of Contents

1. Introduction

IPv6 Rapid Deployment (6rd) as defined in RFC 5969 focuses on rapid deployment of IPv6 by an access service provider which has difficulty deploying native IPv6. This document describes how one service provider in Japan, Sakura Interent, Inc., not for a large residential deployment, but for a large data center network.

While the protocol mechanism of 6rd is unchanged, the deployment model varies a bit from the classical "residential home access provider" model.

The motivation for using 6rd is very similar to that of the residential case where the service provider would like to offer IPv6 quickly to those users who want it, but without replacing equipment that currently does not support IPv6.

This document is provided as information to the Internet community.

2. Network Architecture

The case study presented here is based on the services provide by Sakura Internet Inc. Sakura Internet provides Internet services through Internet backbones and large data centers.

Sakura offers four types of services:

1. Housing Service, which provides Collocation and Internet Access on 5 urban datacenters (4 in Tokyo, 1 in Osaka)

2. Hosting Service, which provides shared service on the servers.

3. Dedicated Server Service, which provides customer dedicated server with variable OSs.

4. Virtual Private Server Service (VPS), which provides guest operating system on the Kernel-based Virtual Machine (KVM).

At the time of this writing, Sakura serves more than 200 Gpbs of traffic on its backbones, and around 50,000 dedicated servers, Virtual Private Servers, and collocated servers.

Figure.1 describes server-based 6rd in datacenter’s network architecture.

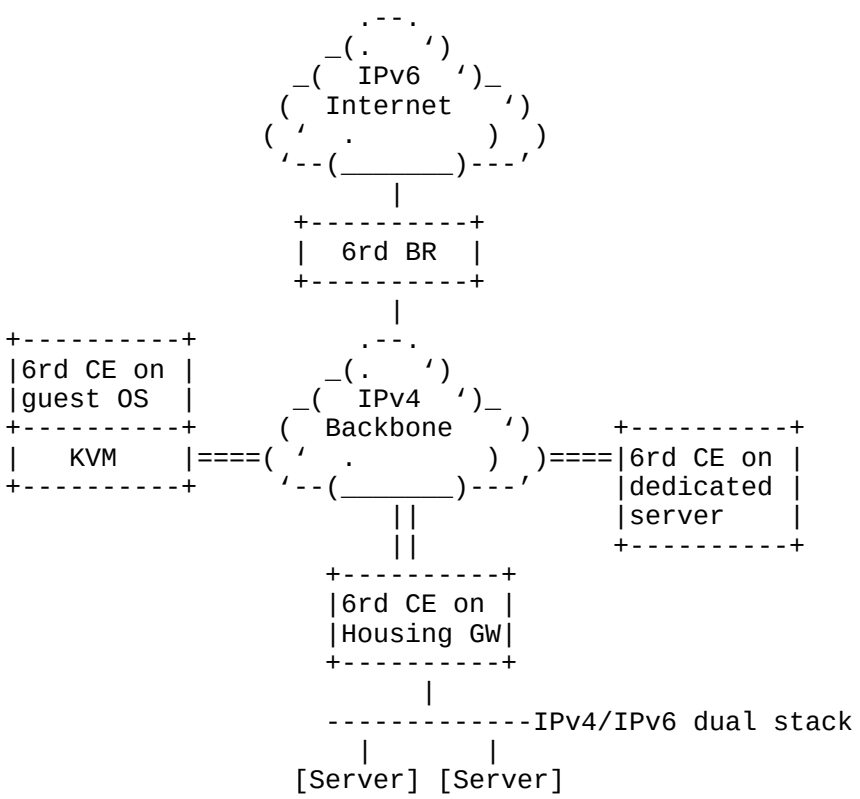


Figure 1

Sakura has deployed commercial 6rd Border Relays, and relies on CE functionality in gateway routers or directly within the operating system of the servers.

In the latter model, there is no need for a CE gateway as the 6rd function is implemented directly in the server operating system itself.

-For Housing users, there are two options. Either the 6rd CE function is performed on the Gateway router itself, or the servers themselves can run 6rd directly.

-For Hosting users, IPv6 service can start by deploying 6rd CE function on the server OS or guest OS on the KVM.

Server administrators can start IPv6 service on demand themselves by using server-based 6rd.

There were some issues when Sakura considers IPv6 deployment on their backbone.

1.Some backbone switches are too old.

IPv6 Switching would be software switching even if IPv4 Switching in hardware. It needs replacement.

2.Some backbone switches required software upgrade.

IPv6 supports on hardware. But software upgraded is needed. In datacenter, there is different requirement on each server, even if the server connected to the same switch. Because the server administrator are completely different. Each server is providing different service to the different customers. So backbones maintenance time negotiation to the customer is very difficult.

To provide native IPv6 service to the existing customer, it needs cost,time and negotiation.

This is the reason why Sakura decided to provide server-based 6rd to the existing customer.

3. 6rd Availability in Server Operating Systems

In particular for the server-initiated case, Sakura relies on 6rd availability in Server operating systems.

Linux kernel has started to support 6rd since 2.6.33. So if Linux based Operating Systems are using 2.6.33 and the later,it can provides server-based 6rd.

FreeBSD and CentOS could not provide 6rd in default, but the patch exist.

Operating Systems	Linux Kernel	Description
Fedora14 and the later	2.6.35 and above	Server-based 6rd ready
Ubuntu 10.10 and the later	2.6.35 and above	Server-based 6rd ready
Debian6.0	2.6.32	Kernel update needs
CentOS5.6	2.6.18	needs [CentOS patch1][CentOS patch2]
FreeBSD8	N/A	needs [BSD patch]

4. Deployment Consideration

4.1. IPv4 compression address

6rd protocol specification is defined on [RFC5969]. Section 4 of [RFC5969] describes o-bit which can compress 32 bit IPv4 address in the 6rd delegated prefix. Linux Kernel also supports this feature.

So customer could get some IPv6 prefixes even if datacenter's prefix is /32.

But [BSD patch] doesn't has the feature of aggregate IPv4 address, therefore datacenter provider has to prepare /32 IPv6 prefix at least in that case.

In Sakura's case, 6rd prefix address using /32, and no compression IPv4 address.Thus the delegated 6rd address length is /64. It is enough address space for server-based 6rd.

4.2. Configuration

Section 7.1 of [RFC5969] describes 6rd CE automatic configuration method such as DHCP,TR-69 and so on.

But server-based 6rd does not needs automatic configuration because the server usually configure IPv4 address statically.

4.3. MTU consideration

Section 9.1 of [RFC5969] describes about Maximum Transmission Unit(MTU) on 6rd tunnel. This guide also applicable for server-based 6rd.

But datacenter's IPv4 network is well-managed and is known by the server administrator. So 6rd CE's tunnel MTU could set be -20 byte from IPv4 MTU.

If the 6rd CE would be TCP server such as WWW, TCP MSS(Maximum Segment Size will be calculated automatically from tunnel MTU.

5. Acknowledgements

The authors thank Hiroki Sato and Masakazu Asama,who made BSD&CentOS patch.

6. IANA Considerations

This document has no actions for IANA.

7. Security Considerations

This document has no security considerations.

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.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

8.2. Informative References

[BSD patch] ""BSD patch"", <http://people.allbsd.org/~hrs/FreeBSD/stf_6rd_20100923-1.diff>.

[CentOS] ""The Community ENTerprise Operating System"", <http://www.centos.org/>.

[CentOS patch1] ""CentOS Kernel patch"", <http://enog.jp/~masakazu/6rd/kernel-2.6.18-238.9.1.el5.6rd.x86_64.rpm>.

[CentOS patch2] ""CentOS iproute patch"", <http://enog.jp/~masakazu/6rd/iproute-2.6.18-11.6rd.x86_64.rpm>.

[Debian] ""Debian -- The Universal Operating System"", <http://www.debian.org/>.

[Fedora] ""Fedora Project Homepage"", <http://fedoraproject.org/>.

[FreeBSD] ""The FreeBSD Project"", <http://www.freebsd.org/>.

[Linux 2.6.33] ""sit: 6rd (IPv6 Rapid Deployment) Support."", <http://kernelnewbies.org/Linux_2_6_33>.

[RFC3849] Huston, G., Lord, A., and P. Smith, "IPv6 Address Prefix Reserved for Documentation", RFC 3849, July 2004.

[RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, January 2010.

[RFC5737] Arkko, J., Cotton, M., and L. Vegoda, "IPv4 Address Blocks Reserved for Documentation", RFC 5737, January 2010.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.

[Ubuntu] ""Ubuntu Homepage"", <http://www.ubuntu.com/>.

Appendix A. Additional Stuff

A.1. OS configuration

A.1.1. Network Topology&Parameters

Describes configuration of each on OS,for reference.

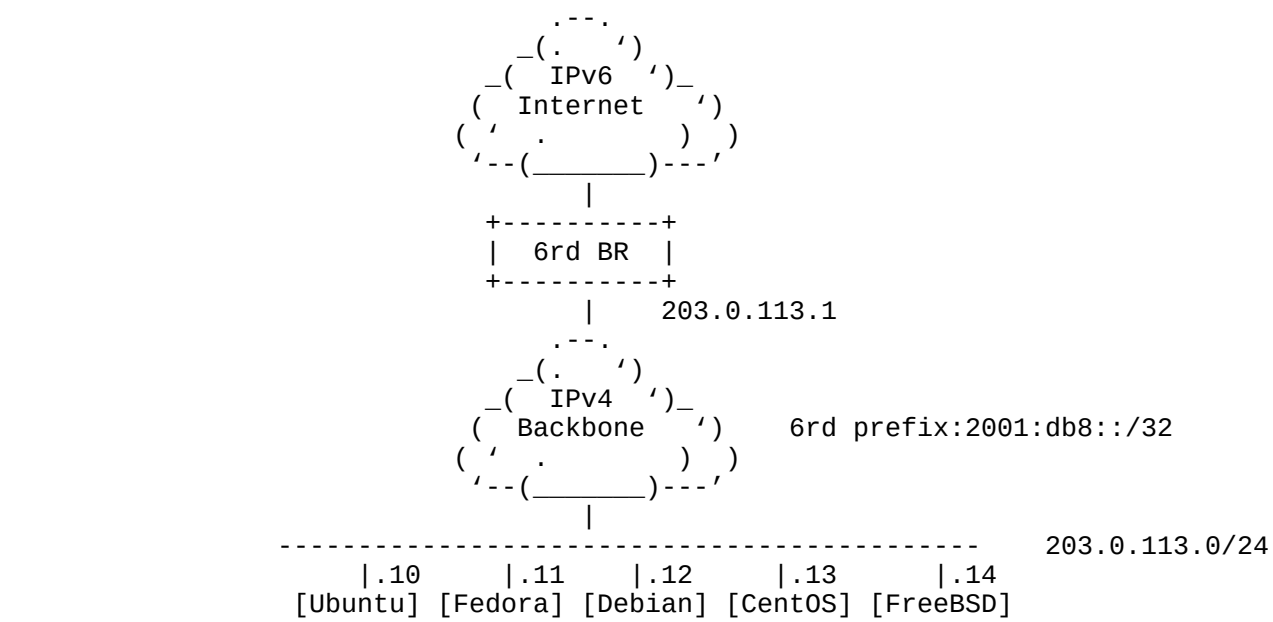


Figure 2

common parameter

BR IPv4 address	6rd prefix	IPv4MaskLen
203.0.113.1	2001:db8::/32	0

individual parameter

OS	IPv4 address	6rd delegated prefix
[Ubuntu]	203.0.113.10	2001:db8:cb00:710a::/64
[Fedora]	203.0.113.11	2001:db8:cb00:710b::/64
[Debian]	203.0.113.12	2001:db8:cb00:710c::/64
[CentOS]	203.0.113.13	2001:db8:cb00:710d::/64
[FreeBSD]	203.0.113.14	2001:db8:cb00:710e::/64

A.1.2. configuration procedure

A.1.2.1. Ubuntu

-modify "/etc/network/interfaces"

```
# vi /etc/network/interfaces
auto tun6rd
iface tun6rd inet6 v4tunnel
  address 2001:db8:cb00:710a::1
  netmask 32
  local 203.0.113.10
  endpoint any
  gateway ::203.0.113.1
  ttl 64
  up ip tunnel 6rd dev tun6rd 6rd-prefix 2001:db8::/32
  up ip link set mtu 1280 dev tun6rd
```

-reboot

A.1.2.2. Fedora

-make "/etc/sysconfig/network-scripts/ifcfg-sit1"

```
# vi /etc/sysconfig/network-scripts/ifcfg-sit1
DEVICE=sit1
IPV6INIT=yes
IPV6_MTU=1280
IPV6_DEFAULTGW=::203.0.113.1
IPV6TUNNELIPV4=any
IPV6TUNNELIPV4LOCAL=203.0.113.11
IPV6ADDR=2001:db8:cb00:710b::1/32
```

-modify "/etc/rc.local"

```
# vi /etc/rc.local
ip tunnel 6rd dev sit1 6rd-prefix 2001:db8::/32
```

-reboot

A.1.2.3. Debian

The latest version of Debian is 6.0.Debian6.0’s kernel is 2.6.32.So it is required upgrade kernel.

-modify "/etc/apt/sources.list"

```
   # vi /etc/apt/sources.list
   deb http://ftp.jp.debian.org/debian experimental main
   deb-src http://ftp.jp.debian.org/debian experimental main

   -upgrade kernel

   # apt-get update
   # apt-get -t experimental install linux-image-2.6.38-rc6-amd64

   -reboot

   -modify "/etc/network/interfaces"

   # vi /etc/network/interfaces
   auto tun6rd
   iface tun6rd inet6 v4tunnel
     address 2001:db8:cb00:710c::1
     netmask 32
     local 203.0.113.12
     endpoint any
     gateway ::203.0.113.1
     ttl 64
     up ip tunnel 6rd dev tun6rd 6rd-prefix 2001:db8::/32
     up ip link set mtu 1280 dev tun6rd

   -reboot
```

A.1.2.4. CentOS

The latest version of CentOS is 5.5.CentOS5.5’s kernel and iproute package does not supported 6rd.So it is required patch.

```
   -download package

# wget http://enog.jp/~masakazu/6rd/kernel-2.6.18-238.9.1.el5.6rd.x86_64.rpm
# wget http://enog.jp/~masakazu/6rd/iproute-2.6.18-11.6rd.x86_64.rpm

   -install package

   # rpm -ivh kernel-2.6.18-238.9.1.el5.6rd.x86_64.rpm
   # rpm -Uvh iproute-2.6.18-11.6rd.x86_64.rpm

   -modify "/etc/yum.conf"

   # vi /etc/yum.conf
   exclude=kernel*,iproute

   -modify "/etc/sysconfig/network-scripts/ifcfg-sit1"
```

```
   # vi /etc/sysconfig/network-scripts/ifcfg-sit1
   DEVICE=sit1
   IPV6INIT=yes
   IPV6_MTU=1280
   IPV6_DEFAULTGW=::203.0.113.1
   IPV6TUNNELIPV4=any
   IPV6TUNNELIPV4LOCAL=203.0.113.13
   IPV6ADDR=2001:db8:cb00:710d::1/32

   modify "/etc/rc.local"

   # vi /etc/rc.local
   ip tunnel 6rd dev sit1 6rd-prefix 2001:db8::/32


   -reboot
```

A.1.2.5. FreeBSD

```
   FreeBSD does not support 6rd yet.But the patch exists.

   -download patch

   # cd /root
   # fetch http://people.allbsd.org/~hrs/FreeBSD/stf_6rd_20100923-1.diff

   -apply patch

   # cd /usr/src
   # patch -p0 < /root/stf_6rd_20100923-1.diff


   -kernel module compile and install

   # cd sys/modules/if_stf/
   # make
   # make install

   -install manual

   # cd /usr/src/share/man/
   # make
   # make install

   -modify "/etc/rc.conf"
```

```
# vi /etc/rc.conf
ipv6_enable="YES"
cloned_interfaces="stf0"
ipv6_ifconfig_stf0="2001:db8:cb00:710e::1/32"
ipv6_defaultrouter="2001:db8:cb00:7101::1"

-reboot
```

A.2. OS Proportion on Sakura's VPS

The data of OS proportion on Sakura's VPS.

All of OSs could server-based 6rd.

Operating Systems	Proportion[%]
Ubuntu	31
Fedora	6
Debian	13
CentOS	39
FreeBSD	11

Authors' Addresses

Shishio Tsuchiya (editor)
Cisco Systems
Shinjuku Mitsui Building, 2-1-1, Nishi-Shinjuku
Shinjuku-Ku, Tokyo 163-0409
Japan

Phone: +81 3 6434 6543
Email: shtsuchi@cisco.com

Mark Townsley
Cisco Systems
L'Atlantis, 11, Rue Camille Desmoulins ISSY LES MOULINEAUX
ILE DE FRANCE 92782
FRANCE

Phone: +33 15 804 3483
Email: mark@townsley.net

Shuichi Ohkubo
Sakura Internet
33F Sumitomo fudosan Nishi shinjuku Bldg.,7-20-1 Nishi shinjuku
Shinjuku-Ku, Tokyo 160-0023
Japan

Phone: +81 3 5332 7070
Email: ohkubo@sakura.ad.jp

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 10, 2011

B. Sarikaya
Huawei USA
June 8, 2011

Multicast Support for 6rd
draft-sarikaya-softwire-6rdmulticast-01.txt

Abstract

This memo specifies modifications required to 6rd so that both IPv6 hosts can receive multicast data from IPv6 servers. The protocol is based on proxying MLD at the 6rd Customer Edge and then tunneling MLD messages to 6rd Border Relays where IPv6 multicast routing is supported. Multicast data received at 6rd Border Relay is tunneled to 6rd Customer Edge node and then delivered to the hosts. We show that IPv4 multicast and IGMP can be supported in a similar way.

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 10, 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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

With IPv4 address depletion on the horizon, many techniques are being standardized for IPv6 migration including 6rd [RFC5969]. 6rd enables IPv6 hosts to communicate with external hosts using IPv4 only legacy ISP network. 6rd Customer Edge (CE) device's LAN side is dual stack and WAN side is IPv4 only. CE tunnels IPv6 packets received from the LAN side to 6rd Border Relays (BR) after encapsulating IPv6 packet in an IPv4 packet. BRs have anycast IPv4 addresses and receive encapsulated packets from CEs over a virtual interface. 6rd operation is stateless. Packets are received/ sent independent of each other and no state needs to be maintained.

6rd as defined in [RFC5969] and [RFC5569] is unicast only. It does not support multicast. In this document we specify how multicast from home IPv6 users can be supported in 6rd. We also show how IPv4 multicast can be supported for home IPv4 users. Both solutions use IPv6 and IPv4 multicast addressing and do not require any new multicast address prefixes such as IPv4-embedded IPv6 multicast addresses to be allocated.

2. Terminology

This document uses the terminology defined in [RFC5969], [RFC5569], [RFC3810] and [RFC3376].

3. Requirements

This section states requirements on 6rd multicast support protocol.

6rd CE MUST support MLD Proxy as defined in [RFC4605]. 6rd CE MAY support IGMP Proxy.

6rd BR MUST support MLD Querrier. 6rd CE MAY support IGMP Querrier.

Both any source multicast (ASM) and source specific multicast (SSM) MUST be supported.

4. Architecture

In 6rd, there are hosts (possibly IPv4/ IPv6 dual stack) served by 6rd Customer Edge device. CE is dual stack facing the hosts and IPv4 only facing the network or WAN side. At the boundary of the network there is 6rd Border Relay. BR receives IPv6 packets tunneled in IPv4 from CE and decapsulates them and sends them out to IPv6 network.

In order to support multicast CE implements MLD Proxy function [RFC4605]. IPv6 hosts send their join requests (MLD Membership Report messages) to CE. CE as a proxy sends aggregated Report messages upstream towards BR.

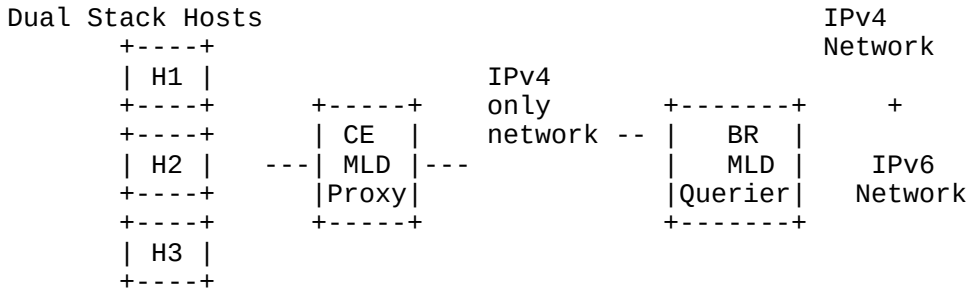


Figure 1: Architecture of 6rd Multicast Protocol

BR is the default multicast querier for CE. BR implements multicast router function or it could be another MLD proxy.

All the elements of 6rd multicast support system are shown in Figure 1.

5. 6rd Multicast Operation

In this section we specify how the host can subscribe and receive IPv6 multicast data from IPv6 content providers based on the architecture defined in Section 4.

The hosts will send their subscription requests for IPv6 multicast groups upstream to the default router, i.e. Costumer Edge device. After subscribing the group, the host can receive multicast data from the CE. The host implements MLD protocol’s host part.

Customer Edge device is MLD Proxy. After receiving the first MLD Report message requesting subscription to an IPv6 multicast group, CE establishes a tunnel interface with a Border Relay. The tunnel is IPv4 based but it will carry IP traffic, MLD messages back and forth and IPv6 multicast data messages downstream. This is similar to [RFC6224] but the operation is much simpler. In 6rd environment there is no requirement to handle host mobility. CE does not have to keep more than one tunnel interfaces, a single interface is

sufficient. MLD Proxy at the CE does not have to have more than one proxy instances, a single instance is sufficient.

CE is regular MLD proxy and it keeps MLD proxy membership database. CE inserts multicast forwarding state on the incoming interface, and merges state updates into the MLD proxy membership database. CE updates or remove elements from the database as required. CE will then send an aggregated Report via the upstream tunnel to the BR when the membership database changes.

CE answers MLD queries from BR based on the membership database. CE's downstream link follows the traditional multipoint channel forwarding and does not pose any specific problems.

CE receives IPv6 multicast data from the BR tunneled over the tunnel interface. CE decapsulates the packet and then forwards it downstream. Each member host receives the data packet based on Layer 2 multicast interface. No packet duplication is necessary.

Border Relay acts as the as the default multicast querier for all CEs that have established an IPv4 tunnel with it. In order to keep a consistent multicast state between a CE and BR, once a CE is connected it will stay connected until the state becomes empty. After that point, the CE may establish another tunnel to a different BR.

According to aggregated MLD reports received from a CE, BR establishes group/source-specific multicast forwarding states at its corresponding downstream tunnel interfaces. After that, BR maintains or removes the state as required by the aggregated reports received from CE.

At the upstream interface, BR procures for aggregated multicast membership maintenance. Based on the multicast-transparent operations of the CEs, the BR treats its tunnel interfaces as multicast enabled downstream links, serving zero to many listening nodes.

Multicast traffic arriving at the BR is transparently forwarded according to its multicast forwarding information base. Multicast data is first replicated and then forwarded in IPv6-in-IPv4 tunnel from BR to the corresponding CE.

5.1. Tunnel Interface Considerations

IPv6 in IPv4 tunneling is performed as specified in [RFC4213]. Considerations specified in [RFC5969] apply. Packets upstream from CE carry only MLD signaling messages and they are not expected to

fragmentation. However packets downstream, i.e. multicast data to CE may be subject to fragmentation.

5.2. Supporting IPv4 Multicast in 6rd

IPv4 multicast can be supported in a way similar to IPv6 as described in Section 5. 6rd Customer Edge device has IGMP Proxy function. Proxy operation for IGMP [RFC3376] is described in [RFC4605].

CE receives IGMP join requests from the hosts and then sends aggregated IGMP Report messages upstream in an IPv4 in IPv4 tunnel. Tunnel addressing is in IPv4 and is as described in [RFC5969]. Multicast membership database is maintained for all active IPv4 multicast groups the hosts subscribe.

6rd Border Relay is IGMP querier or another IGMP Proxy. It serves all CEs downstream and treats its tunnel interfaces as multicast enabled downstream links, serving zero to many listening nodes. Multicast membership database is maintained based on the aggregated Reports received from downstream tunnel interfaces.

IPv4 multicast data received from the multicast Single Source Multicast or Any Source Multicast sources are replicated according to the multicast membership database and the data packets are tunneled to the CEs that have one of more members of this multicast group.

CEs receive multicast data upstream in the CE-BR tunnel and decapsulate it and then forward the packet downstream. Each member host receive IPv4 multicast data packet from its Layer 2 interface.

6. Solution Based on Layer 2 Multicast Support

In this section we assume that Layer 2 multicast is supported in the network. Layer 2 multicast support is done in order to forward multicast data downstream to the ports of Layer 2 devices, i.e. switches that requested a multicast group instead of flooding the data to all the ports.

In the switches called snooping switches, multicast MAC address based filters are setup which link Layer 2 multicast groups to the egress ports. When an MLD Report message is received, the bridge will setup a multicast filter entry that allows (in case of a join message) or prevents (in case of a leave message) packets to flow the port on which the MLD Report message was received. In terms of IP multicast addresses, the mapping is not unique as 2 ** (112 - 32) IPv6 multicast addresses map to a single Ethernet multicast MAC address. This would be reduced to 32 if allocation policy of using only the

lower 32 bits in 112 bit group ID field of IPv6 multicast address is followed.

Snooping switches maintain a list of multicast routers and the ports on which they are attached called router ports. For this purpose multicast router discovery protocol described in [RFC4286] is used. The switch sends an ICMPv6 Multicast Router Solicitation message and the router sends ICMPv6 Multicast Router Advertisement message in reply.

The main functionality of a snooping switch is to forward multicast data packets based on the filters that are setup, i.e. to those egress ports with multicast groups downstream and also to the router ports.

In a 6rd network the snooping switches MUST detect MLD packets in the tunnel between CE and BR. This requires IPv6 snooping switches to be capable of reading IPv4 protocol field values. A value of 58 indicates that an ICMPv6 packet is encapsulated. A value of 41 indicates that an IPv6 data packet is encapsulated. The fact that MLD packets are ICMPv6 packets complicates the operation snooping switch. The switch needs to look further into the packet to correctly identify an MLD packet.

In case multicast is supported in Layer 2, BR after receiving a multicast data packet does not attempt to replicate the packet. The packet replication is taken care of by the snooping switches. So Layer 2 multicast support avoids packet duplication at BR which could be costly in some cases.

7. Security Considerations

TBD.

8. IANA Considerations

TBD.

9. Acknowledgements

TBD.

10. References

10.1. Normative References

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

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, January 2010.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

[RFC4286] Haberman, B. and J. Martin, "Multicast Router Discovery", RFC 4286, December 2005.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.

[RFC6224] Schmidt, T., Waehlisch, M., and S. Krishnan, "Base Deployment for Multicast Listener Support in Proxy Mobile IPv6 (PMIPv6) Domains", RFC 6224, April 2011.

10.2. Informative references

Author's Address

Behcet Sarikaya
Huawei USA
5340 Legacy Dr. Building 175
Plano, TX 75074

Phone:
Email: sarikaya@ieee.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 10, 2011

B. Sarikaya
Huawei USA
June 8, 2011

Multicast Support for Dual Stack Lite
draft-sarikaya-softwire-dslitemulticast-00.txt

Abstract

This memo specifies modifications required to DS-Lite so that both IPv4/ IPv6 hosts can receive multicast data from IPv4/ IPv6 servers.

The DS-Lite solution is based on DS-Lite Basic Bridging BroadBand element (B4) proxying IGMP and then tunneling IGMP messages to DS-Lite Address Family Transition Router element (AFTR). IPv4 multicast data received at AFTR is tunneled to B$ and then delivered to the hosts. IPv6 multicast and MLD can be supported in a similar way.

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 10, 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 to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

With IPv4 address depletion on the horizon, many techniques are being standardized for IPv6 migration including DS-Lite [I-D.ietf-softwire-dual-stack-lite] and 6rd [RFC5969]. DS-Lite enables IPv4 hosts to communicate with external hosts using IPv6 only network and moves the traditional NAT to the network. B4 element's LAN side is dual stack and WAN side is IPv6 only. B4 tunnels IPv4 packets received from the LAN side to AFTR element after encapsulating IPv4 packet in an IPv6 packet. AFTR decapsulates the packet, does a NAT operation and then sends the packet out to IPv4 public internet.

DS-Lite as defined in [I-D.ietf-softwire-dual-stack-lite] is unicast only, it does not support multicast. In this document we specify how multicast from home IPv4 users can be supported in DS-Lite. We also show how IPv6 multicast can be supported for home IPv6 users in DS-Lite.

2. Terminology

This document uses the terminology defined in [I-D.ietf-softwire-dual-stack-lite], [RFC3810] and [RFC3376].

3. Requirements

This section states requirements on DS-Lite multicast support protocol.

DS-Lite B4 MUST support IGMP Proxy as defined in [RFC4605]. DS-Lite B4 MAY support MLD Proxy.

DS-Lite AFTR MUST support IGMP Querrier. DS-Lite AFTR MAY support MLD Querrier.

Both any source multicast (ASM) and source specific multicast (SSM) MUST be supported.

4. Architecture

In DS-Lite, there are hosts (possibly IPv4/ IPv6 dual stack) served by B4 element. B4 is dual stack facing the hosts and IPv6 only facing the network or WAN side. At the boundary of the network there is AFTR. AFTR receives IPv4 packets tunneled in IPv6 from B4 and decapsulates them and sends them out to IPv4 network.

In order to support multicast B4 implements IGMP Proxy function [RFC4605]. IPv4 hosts send their join requests (IGMP Membership Report messages) to B4. B4 as a proxy sends aggregated Report messages upstream towards AFTR.

AFTR is the default multicast querier for B4. AFTR implements multicast router function or it could be another IGMP proxy.

All the elements of DS-Lite multicast support system are shown in Figure 1.

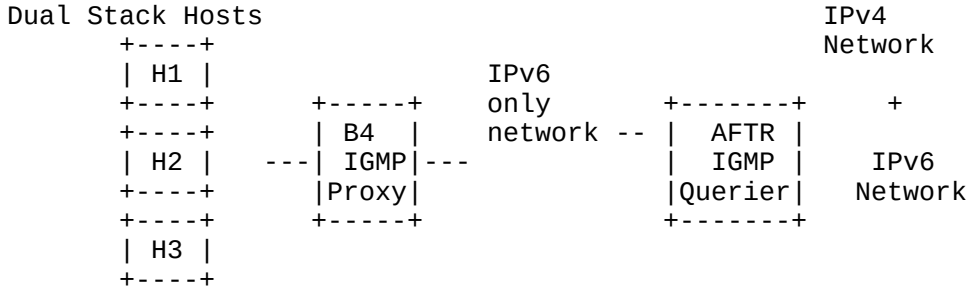


Figure 1: Architecture of DS-Lite Multicast Protocol

5. DS-Lite Multicast Operation

In this section we specify how the host can subscribe and receive IPv4 multicast data from IPv4 content providers based on the architecture defined in Section 4.

The hosts will send their subscription requests for IPv4 multicast groups upstream to the default router, i.e. B4 Element. After subscribing to the group, the host can receive multicast data from the B4. The host implements IGMP protocol’s host part.

B4 Element is IGMP Proxy. After receiving the first IGMP Report message requesting subscription to an IPv4 multicast group, B4 establishes a tunnel interface with a AFTR. The tunnel is IPv6 based but it will carry IPv4 traffic, IGMP messages back and forth and IPv4 multicast data messages downstream. This is similar to [RFC6224] but the operation is much simpler. In DS-Lite environment there is no requirement to handle host mobility. B4 does not have to keep more than one tunnel interfaces, a single interface is sufficient. IGMP Proxy at the B4 does not have to have more than one proxy instances,

a single instance is sufficient.

B4 is regular IGMP proxy and it keeps IGMP proxy membership database. B4 inserts multicast forwarding state on the incoming interface, and merges state updates into the IGMP proxy membership database. B4 updates or removes elements from the database as required. B4 will then send an aggregated Report via the upstream tunnel to the AFTR when the membership database changes.

B4 answers IGMP queries from AFTR based on the membership database. B4's downstream link follows the traditional multipoint channel forwarding and does not pose any specific problems.

B4 receives IPv4 multicast data from the AFTR tunneled over the tunnel interface. B4 decapsulates the packet and then forwards it downstream. Each member host receives the data packet based on Layer 2 multicast interface. No packet duplication is necessary.

AFTR acts as the as the default multicast querier for all B4s that have established an IPv6 tunnel with it. In order to keep a consistent multicast state between a B4 and AFTR, once a B4 is connected it will stay connected until the state becomes empty. After that point, the B4 may continue to use the tunnel for IPv4 unicast traffic.

According to aggregated IGMP reports received from a B4, AFTR establishes group/source-specific multicast forwarding states at its corresponding downstream tunnel interfaces. After that, AFTR maintains or removes the state as required by the aggregated reports received from B4.

At the upstream interface, AFTR procures for aggregated multicast membership maintenance. Based on the multicast-transparent operations of the B4s, the AFTR treats its tunnel interfaces as multicast enabled downstream links, serving zero to many listening nodes.

Multicast traffic arriving at the AFTR is transparently forwarded according to its multicast forwarding information base. Multicast data is first replicated and then forwarded in IPv4-in-IPv6 tunnel from AFTR to the corresponding B4.

5.1. Tunnel Interface Considerations

Legacy IPv4 in IPv6 tunneling is performed as in [RFC2473]. Considerations specified in [I-D.ietf-softwire-dual-stack-lite] apply. Packets upstream from B4 carry only IGMP signaling messages and they are not expected to fragmentation. However packets

downstream, i.e. multicast data to B4 may be subject to fragmentation.

5.2. Supporting IPv6 Multicast in DS-Lite

IPv6 multicast can be supported in a way similar to IPv4 as described in Section 5. B4 Element has MLD Proxy function. Proxy operation for MLD [RFC3810] is described in [RFC4605].

B4 receives MLD join requests from the hosts and then sends aggregated MLD Report messages upstream in an IPv6 in IPv6 tunnel. Tunnel addressing is in IPv6 and is as described in [I-D.ietf-softwire-dual-stack-lite]. Multicast membership database is maintained for all active IPv6 multicast groups the hosts subscribe.

AFTR is MLD querier or another MLD Proxy. It serves all B4s downstream and treats its tunnel interfaces as multicast enabled downstream links, serving zero to many listening nodes. Multicast membership database is maintained based on the aggregated Reports received from downstream tunnel interfaces.

IPv6 multicast data received from the multicast Single Source Multicast or Any Source Multicast sources are replicated according to the multicast membership database and the data packets are tunneled to the B4s that have one or more members of this multicast group.

B4s receive multicast data upstream in the B4-AFTR tunnel and decapsulate it and then forward the packet downstream. Each member host receives IPv6 multicast data packet from its Layer 2 interface.

6. Solution Based on Layer 2 Multicast Support

In this section we assume that Layer 2 multicast is supported in the network. Layer 2 multicast support is done in order to forward multicast data downstream to the ports of Layer 2 devices, i.e. switches that requested a multicast group instead of flooding the data to all the ports.

In the switches called snooping switches, multicast MAC address based filters are setup which link Layer 2 multicast groups to the egress ports. When an IGMP Report message is received, the bridge will setup a multicast filter entry that allows (in case of a join message) or prevents (in case of a leave message) packets to flow the port on which the IGMP Report message was received. In terms of IP multicast addresses the mapping is not unique as 32 IPv4 multicast addresses map to a single Ethernet multicast MAC address.

Snooping switches maintain a list of multicast routers and the ports on which they are attached called router ports. For this purpose multicast router discovery protocol described in [RFC4286] is used. The switch sends an IGMP Multicast Router Solicitation message and the router sends IGMP Multicast Router Advertisement message in reply.

The main functionality of a snooping switch is to forward multicast data packets based on the filters that are setup, i.e. to those egress ports with multicast groups downstream and also to the router ports.

In a DS-Lite network the snooping switches MUST detect IGMP packets in the tunnel between B4 and AFTR. This requires IPv4 snooping switches to be capable of reading IPv6 next header values. A value of 2 indicates that an IGMP packet is encapsulated. A value of 4 indicates that an IPv4 data packet is encapsulated. The switch operates its snooping on these types of packets.

In case multicast is supported in Layer 2, AFTR after receiving a multicast data packet does not attempt to replicate the packet. The packet replication is taken care of by the snooping switches. So Layer 2 multicast support avoids packet duplication at AFTR which could be costly in some cases.

7. IANA Considerations

TBD.

8. Acknowledgements

TBD.

9. References

9.1. Normative References

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

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast

Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

[RFC6224] Schmidt, T., Waehlisch, M., and S. Krishnan, "Base Deployment for Multicast Listener Support in Proxy Mobile IPv6 (PMIPv6) Domains", RFC 6224, April 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-11 (work in progress), May 2011.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.

[RFC4286] Haberman, B. and J. Martin, "Multicast Router Discovery", RFC 4286, December 2005.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

9.2. Informative references

Author's Address

Behcet Sarikaya
Huawei USA
5340 Legacy Drive Building 175
Plano, TX 75074

Phone:
Email: sarikaya@ieee.org

Network Working Group
Internet-Draft
Expires: January 10, 2012

M. Xu
Y. Cui
S. Yang
Tsinghua University
C. Metz
G. Shepherd
Cisco Systems

July 9, 2011

Softwire Mesh Multicast
draft-xu-softwire-mesh-multicast-02

Abstract

The Internet needs support IPv4 and IPv6 packets. Both address families and their attendant protocol suites support multicast of the single-source and any-source varieties. As part of the transition to IPv6, there will be scenarios where a backbone network running one IP address family internally (referred to as internal IP or I-IP) will provide transit services to attached client networks running another IP address family (referred to as external IP or E-IP). It is expected that the I-IP backbone will offer unicast and multicast transit services to the client E-IP networks.

Softwires Mesh is a solution for supporting E-IP unicast and multicast across an I-IP backbone. This document describes the mechanisms for supporting Internet-style multicast across a set of E-IP and I-IP networks supporting softwires mesh.

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 10, 2012.

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction

The Internet needs to support IPv4 and IPv6 packets. Both address families and their attendant protocol suites support multicast of the single-source and any-source varieties. As part of the transition to IPv6, there will be scenarios where a backbone network running one IP address family internally (referred to as internal IP or I-IP) will provide transit services to attached client networks running another IP address family (referred to as external IP or E-IP).

The preferred solution is to leverage the multicast functions, inherent in the I-IP backbone, to efficiently and scalably tunnel encapsulated client E-IP multicast packets inside an I-IP core tree rooted at one or more ingress AFBR nodes and branching out to one or more egress AFBR leaf nodes.

[6] outlines the requirements for the softwires mesh scenario including multicast. It is straightforward to envisage that client E-IP multicast sources and receivers will reside in different client E-IP networks connected to an I-IP backbone network. This requires that the client E-IP source-rooted or shared tree will need to traverse the I-IP backbone network.

One method to accomplish this is to re-use the multicast VPN approach outlined in [10]. MVPN-like schemes can support the softwire mesh scenario and achieve a "many-to-one" mapping between the E-IP client multicast trees and transit core multicast trees. The advantage of this approach is that the number of trees in the I-IP backbone network scales less than linearly with the number of E-IP client trees. Corporate enterprise networks and by extension multicast VPNs have been known to run applications that create a large amount of (S,G) states. Aggregation at the edge contains the (S,G) states that need to be maintained by the network operator supporting the customer VPNs. The disadvantage of this approach is possible inefficient bandwidth and resource utilization if multicast packets are delivered to a receiver AFBR with no attached E-IP receiver.

Internet-style multicast is somewhat different in that the trees tends to be relatively sparse and source-rooted. The need for multicast aggregation at the edge (where many customer multicast trees are mapped into a few or one backbone multicast trees) does not exist and to date has not been identified. Thus the need for a basic or closer alignment with E-IP and I-IP multicast procedures emerges.

A framework on how to support such methods is described in [8]. In this document, a more detailed discussion supporting the "one-to-one" mapping schemes for the IPv6 over IPv4 and IPv4 over IPv6 scenarios will be discussed.

2. Terminology

An example of a softwire mesh network supporting multicast is illustrated in Figure 1. A multicast source S is located in one E-IP client network, while candidate E-IP group receivers are located in the same or different E-IP client networks that all share a common I-IP transit network. When E-IP sources and receivers are not local to each other, they can only communicate with each other through the I-IP core. There may be several E-IP sources for some multicast group residing in different client E-IP networks. In the case of shared trees, the E-IP sources, receivers and RPs might be located in different client E-IP networks. In the simple case the resources of the I-IP core are managed by a single operator although the inter-provider case is not precluded.

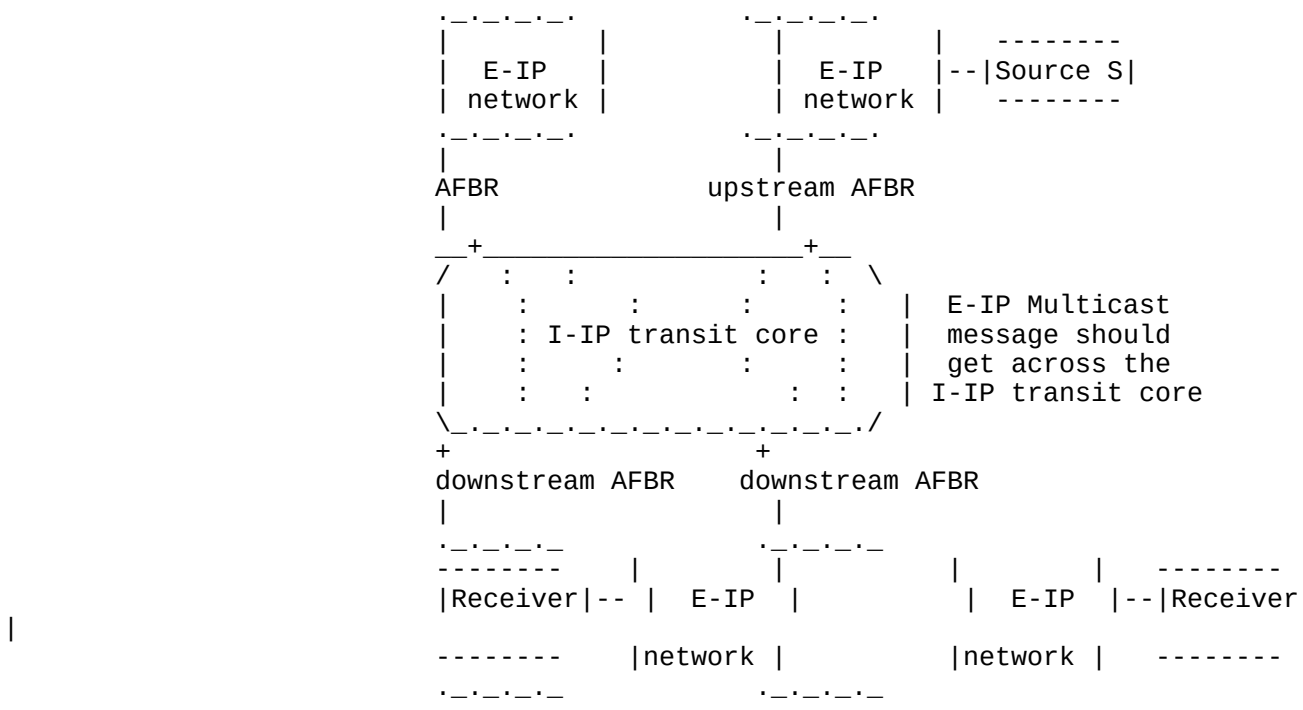


Figure 1: Softwire Mesh Multicast Framework

Terminology used in this document:

o Address Family Border Router (AFBR) - A dual-stack router

interconnecting two or more networks using different IP address families. In the context of softwire mesh multicast, the AFBR runs E-IP and I-IP control planes to maintain E-IP and I-IP multicast states respectively and performs the appropriate encapsulation/ decapsulation of client E-IP multicast packets for transport across the I-IP core. An AFBR will act as a source and/or receiver in an I-IP multicast tree.

o Upstream AFBR: The AFBR router that is located at the upstream of a multicast data flow.

o Downstream AFBR: The AFBR router that is located at the downstream of a multicast data flow.

o I-IP (Internal IP). This refers to the form of IP (i.e., either IPv4 or IPv6) that is supported by the core (or backbone) network. An I-IPv6 core network runs IPv6 and an I-IPv4 core network runs IPv4.

o E-IP (External IP) This refers to the form of IP (i.e. either IPv4 or IPv6) that is supported by the client network(s) attached to the I-IP transit core. An E-IPv6 client network runs IPv6 and an E-IPv4 client network runs IPv4.

o I-IP core tree. A single-source or multi-source distribution tree rooted at one or more AFBR source nodes and branched out to one or more AFBR leaf nodes. An I-IP core Tree is built using standard IP or MPLS multicast signaling protocols operating exclusively inside the I-IP core network. An I-IP core Tree is used to tunnel E-IP multicast packets belonging to E-IP trees across the I-IP core. Another name for an I-IP core Tree is multicast or multipoint softwire.

o E-IP client tree. A single-source or multi-source distribution tree rooted at one or more hosts or routers located inside a client E-IP network and branched out to one or more leaf nodes located in the same or different client E-IP networks.

3. Scenarios of Interest

This section describes the two different scenarios where softwires mesh multicast will apply.

3.1. IPv4-over-IPv6

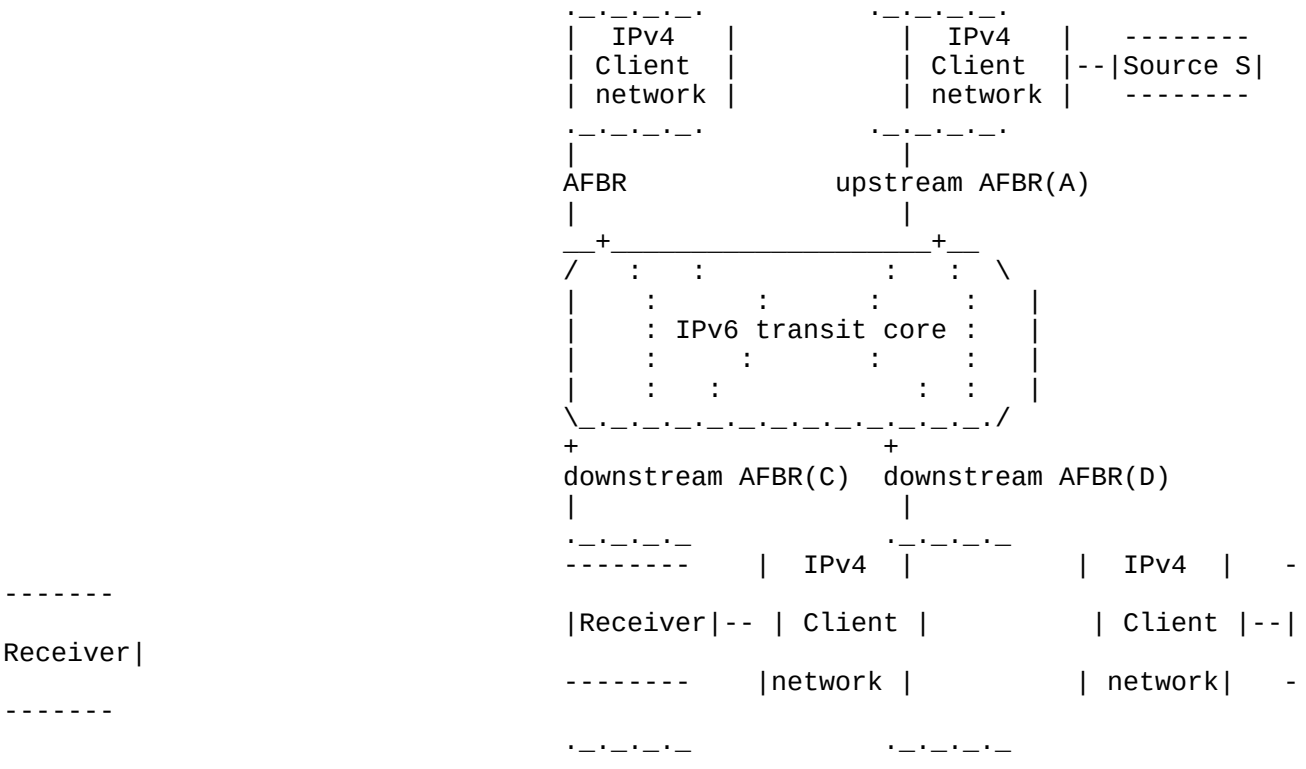


Figure 2: IPv4-over-IPv6 Scenario

In this scenario, the E-IP client networks run IPv4 and I-IP core runs IPv6. This scenario is illustrated in Figure 2.

Because of the much larger IPv6 group address space, it will not be a problem to map individual client E-IPv4 tree to a specific I-IPv6 core tree. This simplifies operations on the AFBR because it becomes possible to algorithmically map an IPv4 group/source address to an IPv6 group/source address and vice-versa.

The IPv4-over-IPv6 scenario is an emerging requirement as network operators build out native IPv6 backbone networks. These networks

naturally support native IPv6 services and applications but it is with near 100% certainty that legacy IPv4 networks handling unicast and multicast will need to be accommodated.

3.2. IPv6-over-IPv4

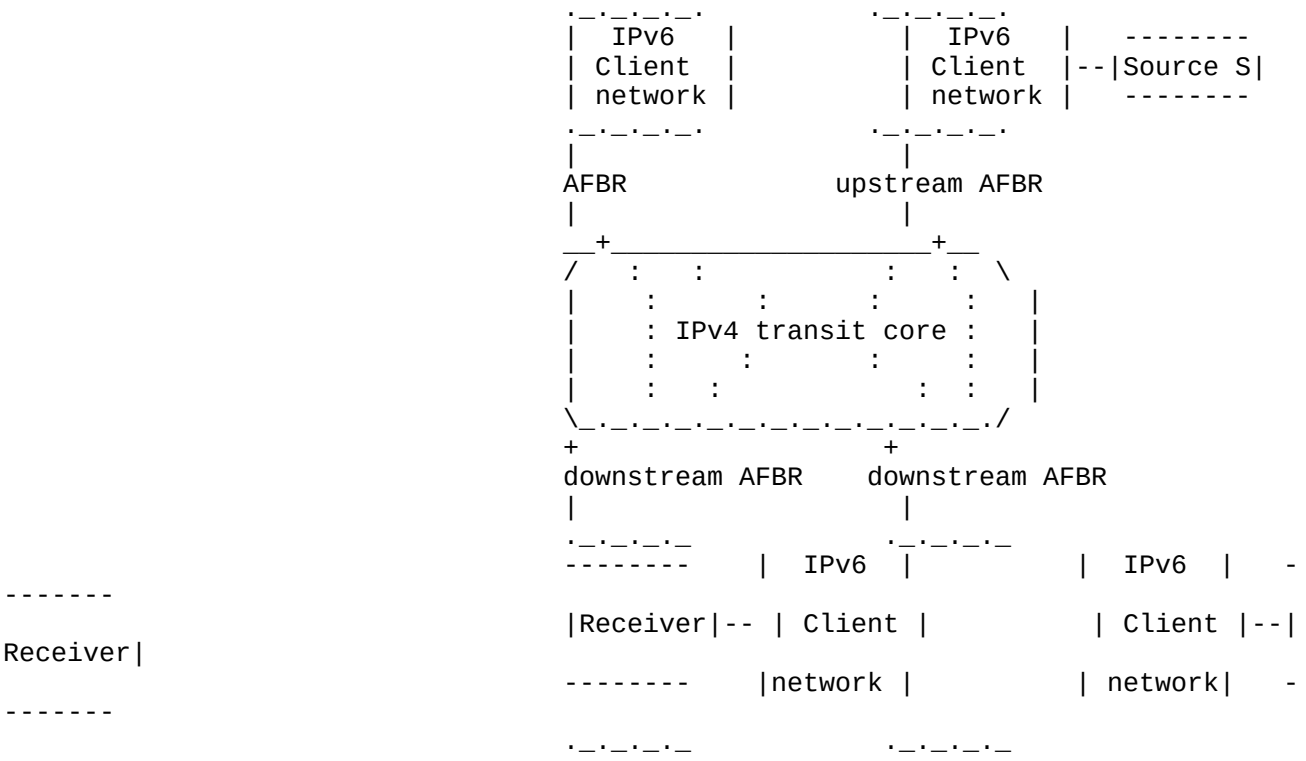


Figure 3: IPv6-over-IPv4 Scenario

In this scenario, the E-IP Client Networks run IPv6 while the I-IP core runs IPv4 and is illustrated in Figure 3.

IPv6 multicast group addresses are longer than IPv4 multicast group addresses. It will not be possible to perform an algorithmic IPv6 - to - IPv4 address mapping without the risk of multiple IPv6 group addresses mapped to the same IPv4 address resulting in unnecessary bandwidth and resource consumption. Therefore additional efforts will be required to ensure that client E-IPv6 multicast packets can be injected into the correct I-IPv4 multicast trees at the AFBRs. This clear mismatch in IPv6 and IPv4 group address lengths means that it will not be possible to perform a one-to-one mapping between IPv6

and IPv4 group addresses unless the IPv6 group address is scoped.

As mentioned earlier this scenario is common in the MVPN environment. As native IPv6 deployments and multicast applications emerge from the outer reaches of the greater public IPv4 Internet, it is envisaged that the IPv6 over IPv4 softwire mesh multicast scenario will be a necessary feature supported by network operators.

4. IPv4-over-IPv6

4.1. Mechanism

Routers in the client E-IPv4 networks contain routes to all other client E-IPv4 networks. Through the set of known and deployed mechanisms, E-IPv4 hosts and routers have discovered or learned of (S,G) or (*,G) IPv4 addresses. Any I-IP multicast state instantiated in the core is referred to as (S’,G’) or (*,G’) and is of course separated from E-IP multicast state.

Suppose a downstream AFBR receives an E-IPv4 PIM Join/Prune message from the E-IPv4 network for either an (S,G) tree or a (*,G) tree. The AFBR can translate the E-IPv4 PIM message into an I-IPv6 PIM message with the latter being directed towards I-IP IPv6 address of the upstream AFBR. When the I-IPv6 PIM message arrives at the upstream AFBR, it should be translated back into an E-IPv4 PIM message. The result of these actions is the construction of E-IPv4 trees and a corresponding I-IP tree in the I-IP network.

In this case it is incumbent upon the AFBR routers to perform PIM message conversions in the control plane and IP group address conversions or mappings in the data plane. It becomes possible to devise an algorithmic one-to-one IPv4-to-IPv6 address mapping at AFBRs.

4.2. Source Address Mapping

There are two kinds of multicast --- ASM and SSM. It’s possible for I-IP network and E-IP network to support different kinds of multicast, and the source address translation rules may vary a lot. There are four scenarios to be discussed in detail:

- E-IP network supports SSM, I-IP network supports SSM
  One possible way to make sure that the translated I-IPv6 PIM message reaches upstream AFBR is to set S’ to a virtual IPv6 address that leads to the upstream AFBR. Figure 4 is the recommended address format based on [9]:

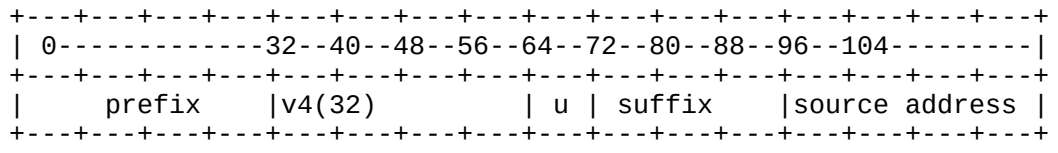


Figure 4: IPv4-Embedded IPv6 Virtual Source Address Format

In this address format, the "prefix" field contains a "Well-Known" prefix or a ISP-defined prefix. An existing "Well-Known" prefix is 64:ff9b, which is defined in [9]; "v4" field is the IP address of one of upstream AFBR's E-IPv4 interface; "u" field is defined in [4], and MUST be set to zero; "suffix" field is reserved for future extensions and SHOULD be set to zero; "source address" field stores the original S.
To make it feasible, the /32 prefix must be known to every AFBR, and AFBRs should not only announce the /96 prefixes of S' to the I-IPv6 network, but also announce the IP addresses of upstream AFBRs' E-IPv4 interface presented in the "v4" field to other AFBRs by MPBGP. In this way, when a downstream AFBR receives a (S,G) message, it can translate it into (S',G') by looking up the IP address of the corresponding AFBR's E-IPv4 interface. Since S' is globally unique and the /96 prefix of S' is known to every router in I-IPv6 network, the translated message will eventually arrive at the corresponding upstream AFBR, and the upstream AFBR can translate the message back to (S,G).

- o E-IP network supports SSM, I-IP network supports ASM
  Since any network that supports ASM should also support SSM, we can construct a SSM tree in I-IP network. The operation in this scenario is the same as that in the first scenario.

- o E-IP network supports ASM, I-IP network supports SSM
  ASM and SSM have the same PIM message format. The main differences between ASM and SSM are RP and (*,G) messages. To make this scenario feasible, we must be able to translate (*,G) messages into (S',G') messages at downstream AFBRs, and translate it back at upstream AFBRs. Assume RP' is the upstream AFBR that locates between RP and the downstream AFBR. When downstream AFBR receives an E-IPv4 PIM (*,G) message, S' can be generated according to the format specified in Figure 4, with "v4" field setting to the IP address of one of RP's E-IPv4 interface and "source address" field setting to *(the IPv4 address of RP). The translated message will eventually arrive at RP'. RP' checks the "source address" field and find the IPv4 address of RP, so RP' judges that this is originally a (*,G) message, then it translates the message back to (*,G) message and forward it to RP.
  Traveling all the way from sources to the RP, and then back down the shared tree may result in the multicast data packets passing through RP' twice, which brings about undesirable increased latency or bandwidth consumption. For this reason, RP' MAY perform a "cut-through", namely when RP' receives multicast data packets sent from sources to RP, it not only forwards them to RP, but also forwards them directly onto the multicast tree built in the I-IPv6 network. (S,G,rpt) messages should be sent towards RP to avoid reduplication.

- o E-IP network supports ASM, I-IP network supports ASM
  To keep it as simple as possible, we treat I-IP network as SSM and the solution is the same as the third scenario.

4.3. Group Address Mapping

For IPv4-over-IPv6 scenario, a simple algorithmic mapping between IPv4 multicast group addresses and IPv6 group addresses is supported. [11] has already defined an applicable format. Figure 5 is a reminder of the format:

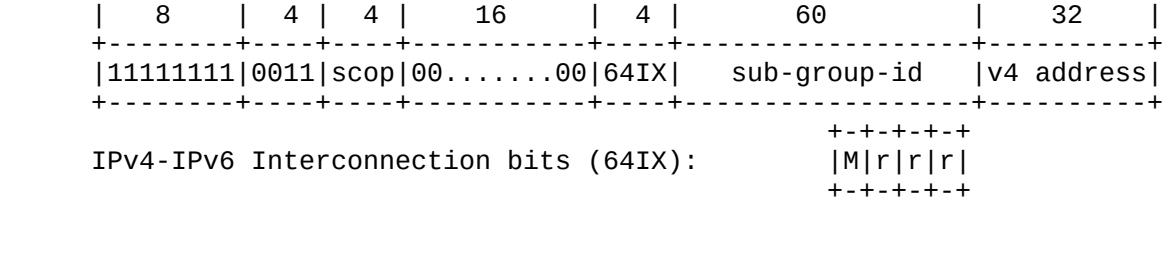


Figure 5: IPv4-Embedded IPv6 Multicast Address Format: SSM Mode

The high order bits of the I-IPv6 address range will be fixed for mapping purposes. With this scheme, each IPv4 multicast address can be mapped into an IPv6 multicast address(with the assigned prefix), and each IPv6 multicast address with the assigned prefix can be mapped into IPv4 multicast address.

4.4. Actions performed by AFBR

The following actions are performed by AFBRs:

- o Receive E-IPv4 PIM messages
  When a downstream AFBR receives an E-IPv4 PIM message, it should check the address family of the next-hop towards the destination. If the address family is IPv4, the AFBR should forward the message without any translation; otherwise it should take the following operation.

- o Translate E-IPv4 PIM messages into I-IPv6 PIM messages
  E-IPv4 PIM message with S(or *) and G is translated into I-IPv6 PIM message with S' and G' following the rules specified above.

- o Transmit I-IPv6 PIM messages
  The downstream AFBR sends the I-IPv6 PIM message to the upstream AFBR. When the upstream AFBR receives this I-IPv6 PIM message, it

checks the prefix of the source address and judges that the message is a translated message, then translates the message back to E-IPv4 PIM message and sends it towards source or RP.

- Process and forward multicast data
  On receiving multicast data from upstream routers, the AFBR looks up its forwarding table to check the IP address of each outgoing interface. If there exists at least one outgoing interface whose IP address family is different from the incoming interface, the AFBR should encapsulate/decapsulate this packet and forward it to the outgoing interface(s), and then forward the data to the other outgoing interfaces without encapsulation/decapsulation.

5. IPv6-over-IPv4

5.1. Mechanism

Routers in the client E-IPv6 networks contain routes to all other client E-IPv6 networks. Through the set of known and deployed mechanisms, E-IPv6 hosts and routers have discovered or learned of (S,G) or (*,G) IPv6 addresses. Any I-IP multicast state instantiated in the core is referred to as (S’,G’) or (*,G’) and is of course separated from E-IP multicast state.

This particular scenario introduces unique challenges. Unlike the IPv4-over-IPv6 scenario, it’s impossible to map all of the IPv6 multicast address space into the IPv4 address space to address the one-to-one Softwire Multicast requirement. To coordinate with the "IPv4-over-IPv6" scenario and keep the solution as simple as possible, one possible solution to this problem is to limit the scope of the E-IPv6 source addresses for mapping, such as applying a "Well-Known" prefix or a ISP-defined prefix.

5.2. Source Address Mapping

There are two kinds of multicast --- ASM and SSM. It’s possible for I-IP network and E-IP network to support different kind of multicast, and the source address translation rules may vary a lot. There are four scenarios to be discussed in detail:

- E-IP network supports SSM, I-IP network supports SSM
  To make sure that the translated I-IPv4 PIM message reaches the upstream AFBR, we need to set S’ to an IPv4 address that leads to the upstream AFBR. But due to the non-"one-to-one" mapping of E-IPv6 to I-IPv4 unicast address, the upstream AFBR is unable to remap the I-IPv4 source address to the original E-IPv6 source address without any constraints.
  We apply a fixed IPv6 prefix and static mapping to solve this problem. A recommended source address format is defined in [9]. Figure 6 is a reminder of the format:

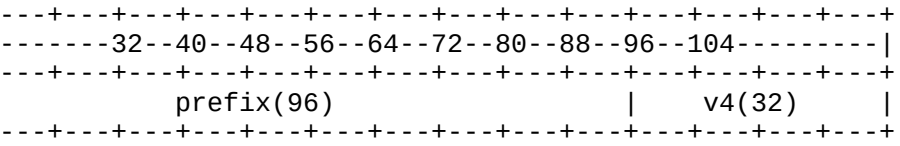


Figure 6: IPv4-Embedded IPv6 Source Address Format

In this address format, the "prefix" field contains a "Well-Known" prefix or a ISP-defined prefix. An existing "Well-Known" prefix is 64:ff9b, which is defined in [9]; "v4" field is the corresponding I-IPv4 source address.
To make it feasible, the /96 prefix must be known to every AFBR, every E-IPv6 address of sources that support mesh multicast MUST follow the format specified in Figure 6, and the corresponding upstream AFBR should announce the I-IPv4 address in "v4" field to the I-IPv4 network. In this way, when a downstream AFBR receives a (S,G) message, it can translate it into (S',G') by simply take off the prefix in S. Since S' is known to every router in I-IPv4 network, the translated message will eventually arrive at the corresponding upstream AFBR, and the upstream AFBR can translate the message back to (S,G) by appending the prefix to S'.

- o E-IP network supports SSM, I-IP network supports ASM
  Since any network that supports ASM should also support SSM, we can construct a SSM tree in I-IP network. The operation in this scenario is the same as that in the first scenario.

- o E-IP network supports ASM, I-IP network supports SSM
  ASM and SSM have the same PIM message format. The main differences between ASM and SSM are RP and (*,G) messages. To make this scenario feasible, we must be able to translate (*,G) messages into (S',G') messages at downstream AFBRs and translate it back at upstream AFBRs. Here, the E-IPv6 address of RP MUST follow the format specified in Figure 6. Assume RP' is the upstream AFBR that locates between RP and the downstream AFBR. When a downstream AFBR receives a (*,G) message, it can translate it into (S',G') by simply take off the prefix in *(the E-IPv6 address of RP). Since S' is known to every router in I-IPv4 network, the translated message will eventually arrive at RP'. RP' knows that S' is the mapped I-IPv4 address of RP, so RP' will translate the message back to (*,G) by appending the prefix to S' and forward it to RP.
  Traveling all the way from sources to the RP, and then back down the shared tree may result in the multicast data packets passing through RP' twice, which brings about undesirable increased latency or bandwidth consumption. For this reason, RP' MAY perform a "cut-through", namely when RP' receives multicast data packets sent from sources to RP, it not only forwards them to RP, but also forwards them directly onto the multicast tree built in the I-IPv6 network. (S,G,rpt) messages should be sent towards RP to avoid reduplication.

- o E-IP network supports ASM, I-IP network supports ASM
  To keep it as simple as possible, we treat I-IP network as SSM and the solution is the same as the third scenario.

5.3. Group Address Mapping

To keep one-to-one group address mapping simple, the group address range of E-IP IPv6 can be reduced in a number of ways to limit the scope of addresses that need to be mapped into the I-IP IPv4 space.

A recommended multicast address format is defined in [11]. The high order bits of the E-IPv6 address range will be fixed for mapping purposes. With this scheme, each IPv4 multicast address can be mapped into an IPv6 multicast address(with the assigned prefix), and each IPv6 multicast address with the assigned prefix can be mapped into IPv4 multicast address.

5.4. Actions performed by AFBR

The following actions are performed by AFBRs

- Receive E-IPv6 PIM messages
  When a downstream AFBR receives an E-IPv6 PIM message, it should check the address family of the upstream router. If the address family is IPv6, the AFBR should not translate this message; otherwise it should take the following operation.

- Translate E-IPv6 PIM messages into I-IPv4 PIM messages
  E-IPv6 PIM message with S(or *) and G is translated into I-IPv4 PIM message with S’ and G’ following the rules specified above.

- Transmit I-IPv4 PIM messages
  The downstream AFBR sends the I-IPv4 PIM message to the upstream AFBR. When the upstream AFBR receives this I-IPv4 PIM message, it checks the source address and judges that the message is a translated message, then translates the message back to E-IPv6 PIM message and sends it towards source or RP.

- Process and forward multicast data
  On receiving multicast data from upstream routers, the AFBR looks up its forwarding table to check the IP address of each outgoing interface. If there exists at least one outgoing interface whose IP address family is different from the incoming interface, the AFBR should encapsulate/decapsulate this packet and forward it to the outgoing interface(s), and then forward the data to the other outgoing interfaces without encapsulation/decapsulation.

6. Security Considerations

The AFBR routers could maintain secure communications through the use of Security Architecture for the Internet Protocol as described in[RFC4301]. But when adopting some schemes that will cause heavy burden on routers, some attacker may use it as a tool for DDoS attack.

7. IANA Considerations

When AFBRs perform address mapping, they should follow some predefined rules, especially the IPv6 prefix for source address mapping should be predefined, so that ingress AFBR and egress AFBR can finish the mapping procedure correctly. The IPv6 prefix for translation can be unified within only the transit core, or within global area. In the later condition, the prefix should be assigned by IANA.

8. References

8.1. Normative References

[1] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[2] Foster, B. and F. Andreasen, "Media Gateway Control Protocol (MGCP) Redirect and Reset Package", RFC 3991, February 2005.

[3] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[4] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[5] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

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

[7] Wijnands, IJ., Boers, A., and E. Rosen, "The Reverse Path Forwarding (RPF) Vector TLV", RFC 5496, March 2009.

[8] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, June 2009.

[9] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

8.2. Informative References

[10] Aggarwal, R., Bandi, S., Cai, Y., Morin, T., Rekhter, Y., Rosen, E., Wijnands, I., and S. Yasukawa, "Multicast in MPLS/ BGP IP VPNs", draft-ietf-l3vpn-2547bis-mcast-10 (work in progress), January 2010.

[11] Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X., and M. Xu, "IPv4-Embedded IPv6 Multicast Address Format", draft-boucadair-behave-64-multicast-address-format-02 (work in progress), June 2011.

Appendix A. Acknowledgements

Wenlong Chen, Xuan Chen, Alain Durand, Yiu Lee, Jacni Qin and Stig Venaas provided useful input into this document.

Authors' Addresses

Mingwei Xu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R. China

Phone: +86-10-6278-5822
Email: xmw@cernet.edu.cn

Yong Cui
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R. China

Phone: +86-10-6278-5822
Email: cuiyong@tsinghua.edu.cn

Shu Yang
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R. China

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

Chris Metz
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

Phone: +1-408-525-3275
Email: chmetz@cisco.com

Greg Shepherd
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

Phone: +1-541-912-9758
Email: shep@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2012

X. Deng
M. Boucadair
France Telecom
C. Zhou
Huawei Technologies
T. Tsou
Huawei Technologies (USA)
G. Bajko
Nokia

July 01, 2011

DS-Lite AFTR NAT Bypass: Co-located B4 and NAT Model
draft-zhou-softwire-b4-nat-02

Abstract

This document describes the behavior of the B4 when co-located with a NAT while the NAT in the AFTR is disabled. The proposed solution is expected to offload the burden on the AFTR, by delegating the NAT to B4.

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 RFC 2119 [RFC2119].

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 2, 2012.

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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

As currently defined in [I-D.ietf-softwire-dual-stack-lite], B4 element SHOULD NOT operate a NAT function because the NAT function will be performed by the AFTR in the service provider's network. To reduce the processing requirement of NAT device at the network side, address and port translation can be made at the customer side, e.g., CPE. For convenience, we call this solution as NAT-Bypass.

This document provides descriptions on the B4 behavior when supporting NAT-Bypass.

2. B4 Behavior

2.1. Provisioning

The provisioning of the B4 element is similar to what is defined in [I-D.ietf-softwire-dual-stack-lite].

2.2. Plain IPv4 Address

A B4 MAY be assigned with a plain IPv4 address.

When a plain, IPv4 address is assigned, the NAT operations are enforced as per current legacy CPEs. The NAT in the AFTR is disabled for that user.

IPv4 datagrams are encapsulated in IPv6 as specified in [I-D.ietf-softwire-dual-stack-lite].

2.3. Restricted IPv4 Address

In the NAT-Bypass solution, the port set is provisioned to B4 through PCP option defined in [I-D.tsou-pcp-natcoord] or specific DHCP options [I-D.bajko-pripaddrassign].

The PCP Server or IPv4 DHCP server may be co-located with the AFTR.

The B4 is responsible for performing NAT and/ALG functions, as well as supporting NAT Traversal mechanisms (e.g., UPnP or NAT-PMP).

2.3.1. Incoming Ports on a given restricted IPv4 address

As described in [I-D.ietf-intarea-shared-addressing-issues], a bulk of incoming ports can be reserved as a centralized resource shared by all subscribers using a given restricted IPv4 address. In order to distribute incoming ports as fair as possible among subscribers

sharing a given restricted IPv4 address, other than allocating a continuous range of ports to each, a solution to distribute bulks of non-continuous ports among subscribers, which also takes port randomization into account, is elaborated in Section 3.1.

2.3.2. Outgoing Packets Processing

Upon receiving an IPv4 packet, the B4 performs NAT using the public IPv4 address and port set assigned to it. Then B4 encapsulates the resulting IPv4 packet into an IPv6 packet, and delivers it through IPv6 connectivity to AFTR which will then decapsulate the encapsulated packet and forward it through IPv4. The destination IPv6 address used for encapsulation should be the AFTR's address.

2.3.3. Incoming Packets Processing

Upon receipt of IPv4-in-IPv6 packet from AFTR, B4 will decapsulate the packet and translate the public IPv4 address to the private IPv4 address. Finally, it delivers the packet to the host using the translated IPv4 address. The source IPv6 address used for encapsulation at AFTR is the AFTR's address, and the destination address is set to the external address of B4.

2.4. Stateless Encapsulation

B4 may implement the stateless encapsulation specified in Section 4.4 of [I-D.ymbk-aplusp].

2.5. Fragmentation and Reassembly

No change to Section 5.3 of [I-D.ietf-softwire-dual-stack-lite].

2.6. DNS

The DNS behavior is the same as described in [I-D.ietf-softwire-dual-stack-lite].

3. Security Considerations

3.1. Port Randomization and non-contiguous port sets allocation mechanism

As port randomization is one protection among others against blind attacks, a simple non-contiguous port sets distribution mechanism is therefore proposed to distribute bulks of non-continuous ports among subscribers, and to enable subscribers operating port randomized NAT.

On every external IPv4 address, according to port set size N, log2(N) bits are randomly choosing by AFTR as subscribers identification bit (s bit) among 1st and 16th bits. Take a sharing ration 1:32 for example, Figure 1 shows an example of 5 random selected bits of s bit.

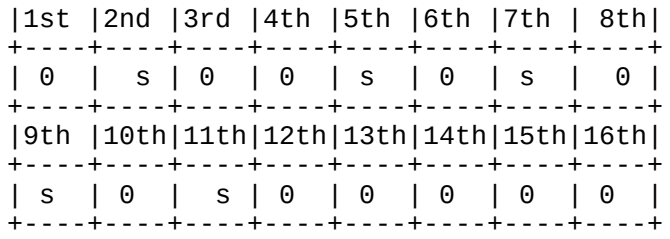


Figure 1: A s bit selection example (on a sharing ration 1:32 address).

Subscriber ID pattern is formed by setting all the s bits to 1 and other trivial bits to 0. Figure 2 illustrates an example of subscriber ID pattern on a sharing ration 1:32 address. Note that the subscriber ID pattern will be different, guaranteed by the random s bit selection, on every restricted IP address no matter whether the sharing ratio varies.

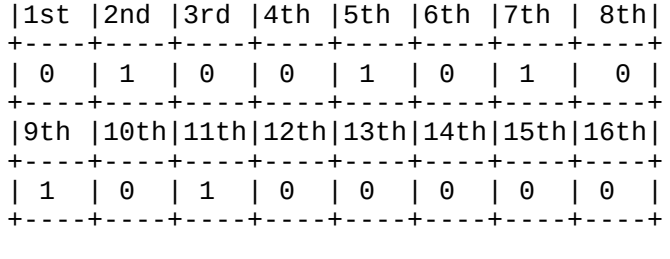


Figure 2: A subscriber ID pattern example (on a sharing ration 1:32 address).

Subscribers ID value is then assigned by setting subscriber ID pattern bits (s bits shown in the following example) according to a customer value and setting other trivial bits to 1.

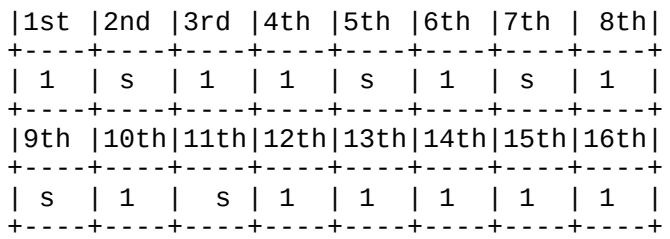


Figure 3: A subscriber ID value example (0# subscriber on this restricted address).

Subscriber ID pattern and subscriber ID value together uniquely defines a non-overlapping port set on a restricted IP address.

Pseudo-code shown in the Figure 4 describe how to use subscriber ID pattern and subscriber ID value to implement a random ephemeral port selection function in a restricted port set.

```
do{
    restricted_next_ephemeral = (random()| customer_ID_pattern)
                                & customer_ID_value;
    if(five-tuple is unique)
    return restricted_next_ephemeral;
}
```

Figure 4: Random ephemeral port selection of restricted port set algorithm.

4. IANA Considerations

None.

5. References

5.1. Normative References

[I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion (Work in progress)", May 2011.

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

5.2. informative References

[I-D.bajko-pripaddrassign]
Bajko, G., Savolainen, T., Boucadair, M., and P. Levis, "Port Restricted IP Address Assignment(Work in progress)", September 2010.

[I-D.ietf-intarea-shared-addressing-issues]
Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing(Work in progress)", March 2011.

[I-D.tsou-pcp-natcoord]
Tsou, T., Zhou, C., Sun, Q., Boucadair, M., and G. Bajko, "Using PCP To Coordinate Between the CGN and Home Gateway Via Port Allocation (Work in progress)", March 2011.

[I-D.ymbk-aplusp]
Bush, R., "The A+P Approach to the IPv4 Address Shortage(Work in progress)", February 2011.

Authors' Addresses

Xiaohong Deng
France Telecom

Email: xiaohong.deng@orange-ftgroup.com

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

Cathy Zhou
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Phone:
Email: cathyzhou@huawei.com

Tina Tsou
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +1 408 330 4424
Email: tena@huawei.com

Gabor Bajko
Nokia

Email: gabor.bajko@nokia.com