

Softwire Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2013

Y. Cui
Tsinghua University
Q. Sun
China Telecom
M. Boucadair
France Telecom
T. Tsou
Huawei Technologies
Y. Lee
Comcast
I. Farrer
Deutsche Telekom AG
February 25, 2013

Lightweight 4over6: An Extension to the DS-Lite Architecture
draft-cui-softwire-b4-translated-ds-lite-11

Abstract

DS-Lite [RFC6333] describes an architecture for transporting IPv4 packets over an IPv6 network. This document specifies an extension to DS-Lite called Lightweight 4over6 which moves the Network Address Translation function from the DS-Lite AFTR to the B4, removing the requirement for a Carrier Grade NAT function in the AFTR. This reduces the amount of centralized state that must be held to a per-subscriber level. In order to delegate the NAPT function and make IPv4 Address sharing possible, port-restricted IPv4 addresses are allocated to the B4s.

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 August 19, 2013.

Copyright Notice

Copyright (c) 2013 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	3
2. Conventions	4
3. Terminology	4
4. Lightweight 4over6 Architecture	5
5. Lightweight B4 Behavior	7
5.1. Lightweight B4 Provisioning	7
5.2. Lightweight B4 Data Plane Behavior	8
6. Lightweight AFTR Behavior	9
6.1. Binding Table Maintenance	9
6.2. lwAFTR Data Plane Behavior	10
7. Provisioning of IPv4 address and Port Set	11
8. ICMP Processing	12
9. Security Considerations	13
10. IANA Considerations	13
11. Author List	13
12. Acknowledgement	16
13. References	16
13.1. Normative References	16
13.2. Informative References	17
Authors' Addresses	18

1. Introduction

Dual-Stack Lite (DS-Lite, [RFC6333]) defines a model for providing IPv4 access over an IPv6 network using two well-known technologies: IP in IP [RFC2473] and Network Address Translation (NAT). The DS-Lite architecture defines two major functional elements as follows:

Basic Bridging BroadBand element: A B4 element is a function implemented on a dual-stack capable node, either a directly connected device or a CPE, that creates a tunnel to an AFTR.

Address Family Transition Router: An AFTR element is the combination of an IPv4-in-IPv6 tunnel endpoint and an IPv4-IPv4 NAT implemented on the same node.

As the AFTR performs the centralized NAT44 function, it dynamically assigns public IPv4 addresses and ports to requesting host's traffic (as described in [RFC3022]). To achieve this, the AFTR must dynamically maintain per-flow state in the form of active NAT sessions. For service providers with a large number of B4 clients, the size and associated costs for scaling the AFTR can quickly become prohibitive. It can also place a large NAT logging overhead upon the service provider in countries where legal requirements mandate this.

This document describes a mechanism called Lightweight 4 over 6 (lw4o6), which provides a solution for these problems. By relocating the NAT functionality from the centralized AFTR to the distributed B4s, a number of benefits can be realised:

- o NAT44 functionality is already widely supported and used in today's CPE devices. Lw4o6 uses this to provide private<->public NAT44, meaning that the service provider does not need a centralized NAT44 function.
- o The amount of state that must be maintained centrally in the AFTR can be reduced from per-flow to per-subscriber. This reduces the amount of resources (memory and processing power) necessary in the AFTR.
- o The reduction of maintained state results in a greatly reduced logging overhead on the service provider.

Operator's IPv6 and IPv4 addressing architectures remain independent of each other. Therefore, flexible IPv4/IPv6 addressing schemes can

be deployed.

Lightweight 4over6 provides a solution for a hub-and-spoke softwire architecture only. It does not offer direct, meshed IPv4 connectivity between subscribers without packets traversing the AFTR. If this type of meshed interconnectivity is required, [I-D.ietf-softwire-map] provides a suitable solution.

The tunneling mechanism remains the same for DS-Lite and Lightweight 4over6. This document describes the changes to DS-Lite that are necessary to implement Lightweight 4over6. These changes mainly concern the configuration parameters and provisioning method necessary for the functional elements.

Lightweight 4over6 features keeping per-subscriber state in the service provider's network. It is categorized as Binding approach in [I-D.bfmk-softwire-unified-cpe] which defines a unified IPv4-in-IPv6 Softwire CPE.

This document is an extended case, which covers address sharing for [I-D.ietf-softwire-public-4over6]. It is also a variant of A+P called Binding Table Mode (see Section 4.4 of [RFC6346]).

This document focuses on architectural considerations and particularly on the expected behavior of the involved functional elements and their interfaces. Deployment-specific issues are discussed in a companion document. As such, discussions about redundancy and provisioning policy are out of scope.

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].

3. Terminology

The document defines the following terms:

Lightweight 4over6 (lw4o6): Lightweight 4over6 is an IPv4-over-IPv6 hub and spoke mechanism, which extends DS-Lite by moving the IPv4 translation (NAPT44) function from the AFTR to the B4.

Lightweight B4 (lwB4): A B4 element (Basic Bridging BroadBand element [RFC6333]), which supports Lightweight 4over6 extensions. An lwB4 is a function implemented on a dual-stack capable node, (either a directly connected device or a CPE), that supports port-restricted IPv4 address allocation, implements NAPT44 functionality and creates a tunnel to an lwAFTR

Lightweight AFTR (lwAFTR): An AFTR element (Address Family Transition Router element [RFC6333]), which supports Lightweight 4over6 extension. An lwAFTR is an IPv4-in-IPv6 tunnel endpoint which maintains per-subscriber address binding only and does not perform a NAPT44 function.

Restricted Port-Set: A non-overlapping range of allowed external ports allocated to the lwB4 to use for NAPT44. Source ports of IPv4 packets sent by the B4 must belong to the assigned port-set. The port set is used for all port aware IP protocols (TCP, UDP, SCTP etc.)

Port-restricted IPv4 Address: A public IPv4 address with a restricted port-set. In Lightweight 4over6, multiple B4s may share the same IPv4 address, however, their port-sets must be non-overlapping.

Throughout the remainder of this document, the terms B4/AFTR should be understood to refer specifically to a DS-Lite implementation. The terms lwB4/lwAFTR refer to a Lightweight 4over6 implementation.

4. Lightweight 4over6 Architecture

The Lightweight 4over6 architecture is functionally similar to DS-Lite. lwB4s and an lwAFTR are connected through an IPv6-enabled network. Both approaches use an IPv4-in-IPv6 encapsulation scheme to deliver IPv4 connectivity services. The following figure shows the data plane with main functional change between DS-Lite and lw4o6:

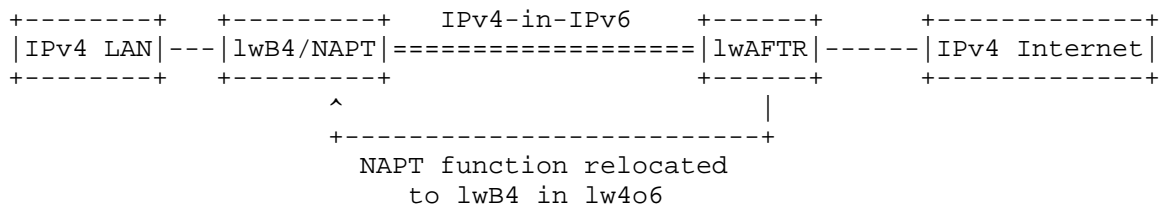


Figure 1 Lightweight 4over6 Data Plane Overview

There are three main components in the Lightweight 4over6 architecture:

- o The lwB4, which performs the NAPT function and encapsulation/de-capsulation IPv4/IPv6.
- o The lwAFTR, which performs the encapsulation/de-capsulation IPv4/IPv6.
- o The provisioning system, which tells the lwB4 which IPv4 address and port set to use.

The lwB4 differs from a regular B4 in that it now performs the NAPT functionality. This means that it needs to be provisioned with the public IPv4 address and port set it is allowed to use. This information is provided through a provisioning mechanism such as DHCP, PCP or TR-69.

The lwAFTR needs to know the binding between the IPv6 address of each subscriber and the IPv4 address and port set allocated to that subscriber. This information is used to perform ingress filtering upstream and encapsulation downstream. Note that this is per-subscriber state as opposed to per-flow state in the regular AFTR case.

The consequence of this architecture is that the information maintained by the provisioning mechanism and the one maintained by the lwAFTR MUST be synchronized (See figure 2). The details of this synchronization depend on the exact provisioning mechanism and will be discussed in a companion draft.

The solution specified in this document allows to assign either a full IPv4 address or shared IPv4 address to requesting CPEs. [I-D.ietf-softwire-public-4over6] provides a mechanism supporting to assign a full IPv4 address only, which could be referred to in this case.

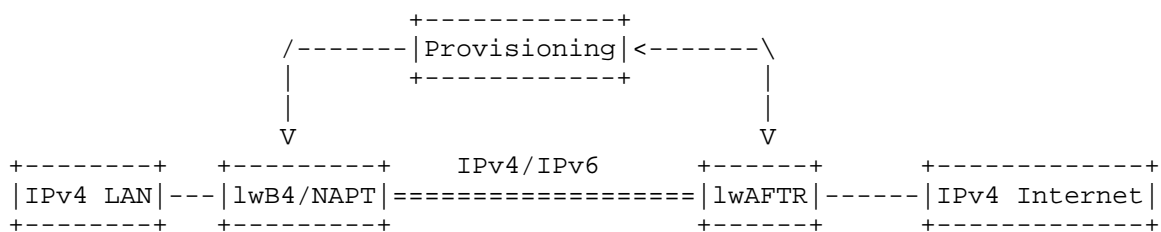


Figure 2 Lightweight 4over6 Provisioning Synchronization

5. Lightweight B4 Behavior

5.1. Lightweight B4 Provisioning

With DS-Lite, the B4 element only needs to be configured with a single DS-Lite specific parameter so that it can set up the softwire (the IPv6 address of the AFTR). Its IPv4 address can be taken from the well-known range 192.0.0.0/29.

In lw4o6, due to the distributed nature of the NAPT function, a number of lw4o6 specific configuration parameters must be provisioned to the lwB4. These are:

- o IPv6 Address for the lwAFTR
- o IPv4 External (Public) Address for NAPT44
- o Restricted port-set to use for NAPT44

An IPv6 address from an assigned prefix is also required for the lwB4 to use as the encapsulation source address for the softwire. Normally, this is the lwB4's globally unique WAN interface address which can be obtained via an IPv6 address allocation procedure such as SLAAC, DHCPv6 or manual configuration.

In the event that the lwB4's encapsulation source address is changed for any reason (such as the DHCPv6 lease expiring), the lwB4's dynamic provisioning process must be re-initiated.

For learning the IPv6 address of the lwAFTR, the lwB4 SHOULD implement the method described in section 5.4 of [RFC6333] and implement the DHCPv6 option defined in [RFC6334]. Other methods of learning this address are also possible.

An lwB4 MUST support dynamic port-restricted IPv4 address provisioning. The potential port set algorithms are described in

[I-D.sun-dhc-port-set-option], and Section 5.1 of [I-D.ietf-softwire-map]. Several different mechanisms can be used for provisioning the lwB4 with its port-restricted IPv4 address such as: DHCPv4, DHCPv6, PCP and PPP. Some alternatives are mentioned in Section 7 of this document.

In this document, lwB4 can be a binding mode CPE. Its provisioning method is RECOMMENDED to follow that is specified in section 3.3 of [I-D.bfmk-softwire-unified-cpe], which will evolve to reflect the consensus from DHC Working Group.

In the event that the lwB4 receives an ICMPv6 error message (type 1, code 5) originating from the lwAFTR, the lwB4 SHOULD interpret this to mean that no matching entry in the lwAFTR's binding table has been found. The lwB4 MAY then re-initiate the dynamic port-restricted provisioning process. The lwB4's re-initiation policy SHOULD be configurable.

The DNS considerations described in Section 5.5 and Section 6.4 of [RFC6333] SHOULD be followed.

5.2. Lightweight B4 Data Plane Behavior

Several sections of [RFC6333] provide background information on the B4's data plane functionality and MUST be implemented by the lwB4 as they are common to both solutions. The relevant sections are:

- | | |
|-----------------------------------|--|
| 5.2. Encapsulation | Covering encapsulation and de-capsulation of tunneled traffic |
| 5.3. Fragmentation and Reassembly | Covering MTU and fragmentation considerations (referencing [RFC2473]) |
| 7.1. Tunneling | Covering tunneling and traffic class mapping between IPv4 and IPv6 (referencing [RFC2473] and [RFC4213]) |

The lwB4 element performs IPv4 address translation (NAPT44) as well as encapsulation and de-capsulation. It runs standard NAPT44 [RFC3022] using the allocated port-restricted address as its external IPv4 address and port numbers.

The lwB4 should behave as is depicted in (2.2) of section 3.2 of [I-D.bfmk-softwire-unified-cpe] when it starts up. The working flow of the lwB4 is illustrated with figure 3.

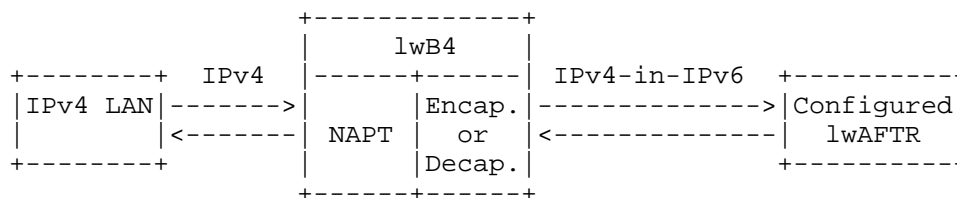


Figure 3 Working Flow of the lwB4

Internally connected hosts source IPv4 packets with an [RFC1918] address. When the lwB4 receives such an IPv4 packet, it performs a NAPT44 function on the source address and port by using the public IPv4 address and a port number from the allocated port-set. Then, it encapsulates the packet with an IPv6 header. The destination IPv6 address is the lwAFTR's IPv6 address and the source IPv6 address is the lwB4's IPv6 tunnel endpoint address. Finally, the lwB4 forwards the encapsulated packet to the configured lwAFTR.

When the lwB4 receives an IPv4-in-IPv6 packet from the lwAFTR, it de-capsulates the IPv4 packet from the IPv6 packet. Then, it performs NAPT44 translation on the destination address and port, based on the available information in its local NAPT44 table.

The lwB4 is responsible for performing ALG functions (e.g., SIP, FTP), and other NAPT traversal mechanisms (e.g., UPnP, NAPT-PMP, manual binding configuration, PCP) for the internal hosts. This requirement is typical for NAPT44 gateways available today.

It is possible that a lwB4 is co-located in a host. In this case, the functions of NAPT44 and encapsulation/de-capsulation are implemented inside the host.

6. Lightweight AFTR Behavior

6.1. Binding Table Maintenance

The lwAFTR maintains an address binding table containing the binding between the lwB4's IPv6 address, the allocated IPv4 address and restricted port-set. Unlike the DS-Lite extended binding table defined in section 6.6 of [RFC6333] which is a 5-tuple NAT table, each entry in the Lightweight 4over6 binding table contains the following 3-tuples:

- o IPv6 Address for a single lwB4

- o Public IPv4 Address
- o Restricted port-set

The entry has two functions: the IPv6 encapsulation of inbound IPv4 packets destined to the lwB4 and the validation of outbound IPv4-in-IPv6 packets received from the lwB4 for de-capsulation.

The lwAFTR does not perform NAT and so does not need session entries.

The lwAFTR MUST synchronize the binding information with the port-restricted address provisioning process. If the lwAFTR does not participate in the port-restricted address provisioning process, the binding MUST be synchronized through other methods (e.g. out-of-band static update).

If the lwAFTR participates in the port-restricted provisioning process, then its binding table MUST be created as part of this process.

For all provisioning processes, the lifetime of binding table entries MUST be synchronized with the lifetime of address allocations.

6.2. lwAFTR Data Plane Behavior

Several sections of [RFC6333] provide background information on the AFTR's data plane functionality and MUST be implemented by the lwAFTR as they are common to both solutions. The relevant sections are:

- | | |
|-----------------------------------|--|
| 6.2. Encapsulation | Covering encapsulation and de-capsulation of tunneled traffic |
| 6.3. Fragmentation and Reassembly | Fragmentation and re-assembly considerations (referencing [RFC2473]) |
| 7.1. Tunneling | Covering tunneling and traffic class mapping between IPv4 and IPv6 (referencing [RFC2473] and [RFC4213]) |

When the lwAFTR receives an IPv4-in-IPv6 packet from an lwB4, it de-capsulates the IPv6 header and verifies the source addresses and port in the binding table. If both the source IPv4 and IPv6 addresses match a single entry in the binding table and the source port in the allowed port-set for that entry, the lwAFTR forwards the packet to

the IPv4 destination.

If no match is found (e.g., no matching IPv4 address entry, port out of range, etc.), the lwAFTR MUST discard or implement a policy (such as redirection) on the packet. An ICMPv6 type 1, code 5 (source address failed ingress/egress policy) error message MAY be sent back to the requesting lwB4. The ICMP policy SHOULD be configurable.

When the lwAFTR receives an inbound IPv4 packet, it uses the IPv4 destination address and port to lookup the destination lwB4's IPv6 address in its binding table. If a match is found, the lwAFTR encapsulates the IPv4 packet. The source is the lwAFTR's IPv6 address and the destination is the lwB4's IPv6 address from the matched entry. Then, the lwAFTR forwards the packet to the lwB4 natively over the IPv6 network.

If no match is found, the lwAFTR MUST discard the packet. An ICMPv4 type 3, code 1 (Destination unreachable, host unreachable) error message MAY be sent back. The ICMP policy SHOULD be configurable.

The lwAFTR MUST support hairpinning of traffic between two lwB4s, by performing de-capsulation and re-encapsulation of packets. The hairpinning policy MUST be configurable.

7. Provisioning of IPv4 address and Port Set

There are several dynamically provisioning protocols for IPv4 address and port set. These protocols MAY be implemented. Some possible alternatives include:

- o DHCP: Extending DHCP protocol MAY be used for the provisioning [I-D.ietf-dhc-dhcpv4-over-ipv6] [I-D.ietf-softwire-map-dhcp].
- o PCP[I-D.ietf-pcp-base]: a lwB4 MAY use [I-D.tsou-pcp-natcoord] to retrieve a restricted IPv4 address and a set of ports.

In a Lightweight 4over6 domain, the same provisioning mechanism MUST be enabled in the lwB4s, the AFTRs and the provisioning server.

DHCP-based provisioning mechanism (DHCPv4/DHCPv6) is RECOMMENDED in this document. The provisioning mechanism for port-restricted IPv4 address will evolve according to the consensus from DHC Working Group.

8. ICMP Processing

ICMP does not work in an address sharing environment without special handling [RFC6269]. Due to the port-set style address sharing, Lightweight 4over6 requires specific ICMP message handling not required by DS-Lite.

The following behavior SHOULD be implemented by the lwAFTR to provide ICMP error handling and basic remote IPv4 service diagnostics for a port restricted CPE: for inbound ICMP messages, the lwAFTR MAY behave in two modes:

Either:

1. Check the ICMP Type field.
2. If the ICMP type is set to 0 or 8 (echo reply or request), then the lwAFTR MUST take the value of the ICMP identifier field as the source port, and use this value to lookup the binding table for an encapsulation destination. If a match is found, the lwAFTR forwards the ICMP packet to the IPv6 address stored in the entry; otherwise it MUST discard the packet.
3. If the ICMP type field is set to any other value, then the lwAFTR MUST use the method described in REQ-3 of [RFC5508] to locate the source port within the transport layer header in ICMP packet's data field. The destination IPv4 address and source port extracted from the ICMP packet are then used to make a lookup in the binding table. If a match is found, it MUST forward the ICMP reply packet to the IPv6 address stored in the entry; otherwise it MUST discard the packet.

Or:

- o Discard all inbound ICMP messages.

The ICMP policy SHOULD be configurable.

The lwB4 SHOULD implement the requirements defined in [RFC5508] for ICMP forwarding. For ICMP echo request packets originating from the private IPv4 network, the lwB4 SHOULD implement the method described in [RFC6346] and use an available port from its port-set as the ICMP Identifier.

For both the lwAFTR and the lwB4, ICMPv6 MUST be handled as described in [RFC2473].

9. Security Considerations

As the port space for a subscriber shrinks due to address sharing, the randomness for the port numbers of the subscriber is decreased significantly. This means it is much easier for an attacker to guess the port number used, which could result in attacks ranging from throughput reduction to broken connections or data corruption.

The port-set for a subscriber can be a set of contiguous ports or non-contiguous ports. Contiguous port-sets do not reduce this threat. However, with non-contiguous port-set (which may be generated in a pseudo-random way [RFC6431]), the randomness of the port number is improved, provided that the attacker is outside the Lightweight 4over6 domain and hence does not know the port-set generation algorithm.

More considerations about IP address sharing are discussed in Section 13 of [RFC6269], which is applicable to this solution.

10. IANA Considerations

This document does not include an IANA request.

11. Author List

The following are extended authors who contributed to the effort:

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: pengwu.thu@gmail.com

Qi Sun
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

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

Xiaohong Deng
France Telecom

Email: xiaohong.deng@orange.com

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

Email: cathyzhou@huawei.com

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

Email: adurand@juniper.net

Reinaldo Penno
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: repenno@cisco.com

Alex Clauberg
Deutsche Telekom AG
GTN-FM4
Landgrabenweg 151
Bonn, CA 53227
Germany

Email: axel.clauberg@telekom.de

Lionel Hoffmann
Bouygues Telecom
TECHNOPOLE
13/15 Avenue du Marechal Juin
Meudon 92360
France

Email: lhoffman@bouyguestelecom.fr

Maoke Chen
FreeBit Co., Ltd.
13F E-space Tower, Maruyama-cho 3-6
Shibuya-ku, Tokyo 150-0044
Japan

Email: fibrib@gmail.com

12. Acknowledgement

The authors would like to thank Ole Troan, Ralph Droms and Suresh Krishnan for their comments and feedback.

This document is a merge of three documents:

[I-D.cui-softwire-b4-translated-ds-lite], [I-D.zhou-softwire-b4-nat] and [I-D.penno-softwire-sdnat].

13. References

13.1. Normative References

- [I-D.bfmk-softwire-unified-cpe]
Boucadair, M. and I. Farrer, "Unified IPv4-in-IPv6 Softwire CPE", draft-bfmk-softwire-unified-cpe-02 (work in progress), January 2013.
- [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.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.
- [RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.
- [RFC5508] Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT Behavioral Requirements for ICMP", BCP 148, RFC 5508, April 2009.
- [RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.
- [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.
- [RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.
- [RFC6431] Boucadair, M., Levis, P., Bajko, G., Savolainen, T., and T. Tsou, "Huawei Port Range Configuration Options for PPP IP Control Protocol (IPCP)", RFC 6431, November 2011.

13.2. Informative References

- [I-D.cui-software-b4-translated-ds-lite]
Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture", draft-cui-software-b4-translated-ds-lite-10 (work in progress), February 2013.
- [I-D.ietf-dhc-dhcpv4-over-ipv6]
Cui, Y., Wu, P., Wu, J., and T. Lemon, "DHCPv4 over IPv6 Transport", draft-ietf-dhc-dhcpv4-over-ipv6-05 (work in progress), September 2012.
- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-29 (work in progress), November 2012.
- [I-D.ietf-software-map]
Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., and T. Murakami, "Mapping of Address and Port with Encapsulation (MAP)", draft-ietf-software-map-04 (work in progress), February 2013.
- [I-D.ietf-software-map-dhcp]
Mrugalski, T., Troan, O., Dec, W., Bao, C., leaf.yeh.sdo@gmail.com, l., and X. Deng, "DHCPv6 Options for Mapping of Address and Port", draft-ietf-software-map-dhcp-03 (work in progress), February 2013.
- [I-D.ietf-software-public-4over6]
Cui, Y., Wu, J., Wu, P., Vautrin, O., and Y. Lee, "Public IPv4 over IPv6 Access Network",

draft-ietf-softwire-public-4over6-04 (work in progress),
October 2012.

[I-D.penno-softwire-sdnat]

Penno, R., Durand, A., Hoffmann, L., and A. Clauberg,
"Stateless DS-Lite", draft-penno-softwire-sdnat-02 (work
in progress), March 2012.

[I-D.sun-dhc-port-set-option]

Sun, Q., Lee, Y., Sun, Q., Bajko, G., and M. Boucadair,
"Dynamic Host Configuration Protocol (DHCP) Option for
Port Set Assignment", draft-sun-dhc-port-set-option-00
(work in progress), October 2012.

[I-D.tsou-pcp-natcoord]

Sun, Q., Boucadair, M., Deng, X., Zhou, C., Tsou, T., and
S. Perreault, "Using PCP To Coordinate Between the CGN and
Home Gateway", draft-tsou-pcp-natcoord-09 (work in
progress), November 2012.

[I-D.zhou-softwire-b4-nat]

Zhou, C., Boucadair, M., and X. Deng, "NAT offload
extension to Dual-Stack lite",
draft-zhou-softwire-b4-nat-04 (work in progress),
October 2011.

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

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

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

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Tina Tsou
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +1-408-330-4424
Email: tena@huawei.com

Yiu L. Lee
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: yiu_lee@cable.comcast.com

Ian Farrer
Deutsche Telekom AG
GTN-FM4, Landgrabenweg 151
Bonn, NRW 53227
Germany

Email: ian.farrer@telekom.de

Internet Engineering Task Force
Internet Draft
Intended status: Informational
Expires: September 2, 2012

X.Deng
M.Boucadair
France Telecom
Y.Lee
Comcast
X.Huang
Q.Zhao
BUPT
March 1, 2012

Implementing A+P in the provider's IPv6-only network
draft-deng-softwire-aplusp-experiment-results-02.txt

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 flavor of A+P is also implemented to verify feasibility of offering non-continuous port sets with A+P approach.

The test results consist of the application compatibility test, UPnP 1.0 extensions and UPnP 1.0 friendly port allocation for A+P, port usage and BitTorrent behaviors with A+P.

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

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 2, 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. Introduction	3
2. Terminology	3
3. Implementation environment	4
3.1. Environment Overview	4
3.2. Implementation and Configuration of A+P	5
3.2.1. IPv4-Embedded IPv6 Address Format For A+P CPE	6
3.2.2. DHCPv6 Configurations	6
3.2.3. Avoiding Fragmentation	6
3.3. Implementing non-continuous Port Sets for A+P	7
3.3.1. Non-continuous Port Sets allocation mechanism	7
3.3.2. IPv4-Embedded IPv6 Address Format for Non-continuous Port Sets A+P CPE	10
3.3.3. Customize a non-continuous Ports Set A+P NAT	11
4. Application Tests and Experiments in A+P Environment	12
4.1. A+P Impacts on Applications	12
4.2. UPnP extension experiment	13
4.2.1. UPnP 1.0 extension	13
4.2.2. UPnP 1.0 friendliness attempts	14
4.3. Port Usage of Applications	16
4.4. BitTorrent Behaviour in A+P	17
5. Security Considerations	18
6. IANA Considerations	18
7. Conclusion	18
8. References	19
8.1. Normative References	19
8.2. Informative References	19
9. Additional Authors	20
10. Acknowledgments	20

1. Introduction

A+P [RFC6346] is a technique to share IPv4 addresses over IPv6-only network without requiring a NAT function in the provider's network. The main idea of A+P is borrowing some bits from the port number in the TCP/UDP header to identify the end point. Those port numbers assigned to the end point will be used by IPv4 applications. A+P can facilitate network migration to IPv6-only while continue to offer IPv4 connectivity to customers by tunneling IPv4 packets over IPv6-only network.

We implemented A+P in a residential ADSL access network, where IPv6-only access network is provided over PPPoE. In this memo, we first describe the implementation environment including A+P IPv6 prefix format and network elements configurations, then we describes the test results. In particular, this memo focuses on the SMAP function implementation specified in [RFC6346].

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

2. Terminology

This memo uses 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

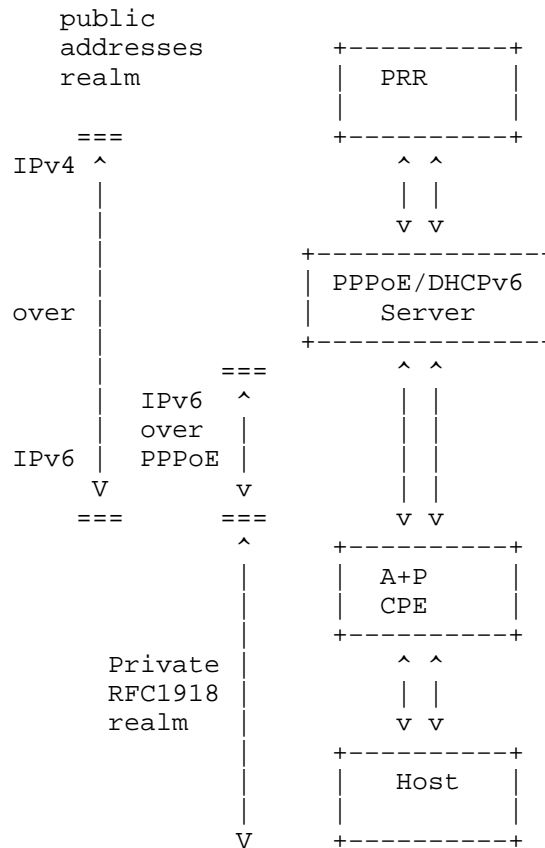


Figure 1 : Implementation Environment

We developed both A+P CPE function and Port Range Router (PRR) function on Linux. A+P CPE function was implemented on Linksys WRT54GS router running OpenWRT 2.6.32. PRR function was implemented on standard Intel based server. Figure 1 shows the high-level network diagram of the test environment.

Figure 2 shows the configuration of A+P CPE. IPv6 prefix was provisioned over PPPoE to CPE by a DHCPv6 server. In addition, it also offered A+P parameters via DHCPv6 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

A+P CPE uses Netfilter framework to implement the port-set restricted NAT. Port set restricted NAT operation was done by iptables rules. After the port restricted NAT operation, IPv4 packets were sent to a TUN interface which was a virtual network interface in Linux. The TUN interface is a virtual interface that performs the IPv4-in-IPv6 function. Using the IPv4-Embedded IPv6 address format defined in section 3.2.1, an IPv4-in-IPv6 function is performed by the TUN interface handler.

PRR bridges the IPv6 access network to the IPv4 Internet. It contains two main functions: 1) IPv4-in-IPv6 encapsulation/decapsulation; Similar to A+P CPE, PRR implementation leveraged the virtual TUN driver handler for IPv4-in-IPv6 function. 2) Destination IPv4 address and layer 4 port based routing function is responsible for routing the IPv4 traffic originated from the IPv4 Internet to the Port Range restricted A+P CPE. The goal of PRR is to deliver the IPv4 packet to the A+P CPE that was assigned with the port number used in the destination port in the layer 4 header. Since PRR delivers the IPv4 packet over IPv4-in-IPv6 tunnel, PRR can embed the IPv4 address and port number in the IPv6 address. The IPv4-Embedded IPv6 address is used to uniquely identify the A+P CPE. Details of how to construct the IPv4-Embedded IPv6 address format is defined in Section 3.2.1.

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

31bits	1bit	32bits	8 bits	16bits	4bits	1bit	1bit	1bit	1bit	32 bits
A+P Prefix	flag 0	Public IPv4 Address	EUI64	port Range	Port Range Size	flag 1	flag 2	flag 3	flag 4	Public IPv4 Address

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 other parties who are also interested in testing A+P solution, we are considering to release this A+P 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] were implemented. These options allow configuring a shared address and a port range using a DHCPv6 option.

3.2.3. Avoiding Fragmentation

Normally the host TCP/IP protocol stack uses TCP protocol stack uses Maximum Segment Size (MSS) option and/or Path Maximum Transmission Unit Discovery (PMTUD) to determine the MTU.

However adding the IPv6 Header and the PPPoE header to the IPv4 packet may exceed the maximum MTU of the wire and consequently results in IP fragmentation.

One solution is to add a rule to iptables on A+P CPE to modify the MSS value in TCP SYN and SYN-ACK. This can be done using command "iptables -t mangle -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --set-mss DESIRED_MSS_VALUE". The DESIRED_MSS_VALUE is set to exclude IPv4 header, TCP header, IPv6 header and PPPoE header length.

3.3. Implementing non-continuous Port Sets for A+P

3.3.1. Non-continuous Port Sets allocation mechanism

[I-D.ietf-intarea-shared-addressing-issues] states that a bulk of incoming ports can be reserved as a centralized resource shared by all subscribers using a given restricted IPv4 address. We could distribute a range of continuous ports to each subscriber. This may create security concerns such as blind attack. An alternative would be to assign a bulk of non-continuous random ports to each subscriber. The following session would describe the implementation of non-continuous port-set.

Note that the non-continuous port-set allocation mechanism described here is just one possible solution to implement non-continuous port provisioning. The implementation itself is to achieve two goals: 1) Proving of feasibility of non-continuous port-set with A+P approach; 2) Evaluating UPnP 1.0 compatibility with non-continuous port-set. Experiment results are provided in Section 4.2.2. Given a port-set size N , $\log_2(N)$ bits are randomly chosen as subscribers identification bits (S-bit). S-bit must be chosen between 1st and 16th bits. For example: if sharing ration is 1:32, each subscriber will have five S-bits. Figure 4 shows an example of 5 S-bits (2nd, 5th, 7th, 9th and 11th) for a subscriber.

Subscriber ID pattern is formed by setting all the S-bits to 1 and other trivial bits to 0. Figure 5 illustrates an example of subscriber ID pattern based on S-bits example in Figure 4.

Note that the subscriber ID pattern must be identical for each subscriber that shares the same IPv4 address.

Subscribers ID value is assigned by setting subscriber ID pattern bits (s bits shown in figure 4) to a unique customer value to identify each customer 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.

1st	2nd	3rd	4th	5th	6th	7th	8th
0	s	0	0	s	0	s	0
9th	10th	11th	12th	13th	14th	15th	16th
s	0	s	0	0	0	0	0

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

1st	2nd	3rd	4th	5th	6th	7th	8th
0	1	0	0	1	0	1	0
9th	10th	11th	12th	13th	14th	15th	16th
1	0	1	0	0	0	0	0

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

1st	2nd	3rd	4th	5th	6th	7th	8th
1	0	1	1	0	1	0	1

9th	10th	11th	12th	13th	14th	15th	16th
0	1	0	1	1	1	1	1

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

Subscriber ID pattern and subscriber ID value together uniquely define 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 Non-continuous Port Sets A+P CPE

31bits	1bit	32bits	8bits	16bits	4bits	1bit	1bit	1bit	1bit	32bits
A+P Prefix	flag 0	Public IPv4 Address	EUI64	SID_ Value	Reser -ved	flag 1	flag 2	flag 3	flag 4	Public IPv4 Address

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?

To support non-continuous port-set, PRR maintains a mapping table which contains the pairs 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.

$$\text{SID_Value} = \text{Destination_Port} \mid (\sim \text{SID_Pattern});$$

Figure 9 :PRR calculates SID Value

3.3.3. Customize a non-continuous Ports Set A+P NAT

On a Linux kernel 2.6.32.36, only one line of linux kernel code was Changed to implement this feature. Figure 10 shows the change. Figure 11 show the IPTables commands required in the PRR. The beginning of port range changed to SID_Value and the ending of the port range changed to SID_Pattern.

```
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 non-continuousport 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 non-continuousports set A+P NAT

4. Application Tests and Experiments in A+P Environment

A set of well-known applications was tested. The tests compared A+P over IPv6 and simple A+P without encapsulation on a pain IPv4 network. The test results showed that both share the same impacts [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: A+P impacts on applications

P2P (Peer-to-Peer) applications using specific port for inbound connection are likely to fail, because the specific ports may not be available for that A+P subscriber. 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

4.2.1. UPnP 1.0 extension

To make P2P application work properly with port restricted NAT , we have designed extensions including new variables, new error codes 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.

4.2.2. UPnP 1.0 friendliness attempts

Application	Behaviors
Microtorrent v2.2 (also known as uTorrent)	call GetSpecificPortMapping by incremental by 1 each time, until find an external port available, and then call AddPortMapping, or return error after five failures
Emule v0.50a	call AddPortMapping, after finding the external port not available return error
Azureus v4.6.0.2	call AddPortMapping, after finding the external port not available, try the same port 5 more times by call AddPortMapping, then return error
Shareazav2.2.5.7	call GetSpecificPortMapping, after finding the external port not available, return error without issuing AddPortMapping

Figure 16 UPnP 1.0 applications behaviors of asking for an external port

The Behaviors test results in the previous figure shows that if a request of external port failed, some UPnP 1.0 applications, namely Microtorrent v2.2 and Azureus v4.6.0.2 attempt to issue (at most) 4 more times request until succeed. With each external port request attempts, the desired external port is incremented by 1 of the previous requested external port.

Hence, allocating port sets in a way that each A+P subscriber has sub sets interval less than 5 would make some UPnP 1.0 applications succeed in 5 times retrying. For example, In case a Subscriber ID Pattern 0x02 that makes 2 customers sharing one IPv4 address, and customer 1 have the available ports
{ 0,1 | 4,5 | 8,9 |12,13|....} while customer 2 have the available ports:
{ 2,3 | 6,7 | 10,11|14,15|....}

Microtorrent v2.2 and Azureus v4.6.0.2 would be compatible with port restriction feature of A+P.

IGD:1 is known to be broken in shared address environment [RFC6269]; IGD:2 mitigates the issues encountered in IGD:1. The efforts, documented in section 4.2, were attempts before standardization of IGD:2.

4.3. Port Usage of Applications

Port consumptions of applications not only impact the deployment factor (i.e., port range size) for A+P 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 A+P CPE for NAT session usage, and store A+P NAT statistics in a Mysql database for further analysis of application behaviours 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 17: Port usage of applications

4.4. BitTorrent Behaviour in A+P

[draft-boucadair-behave-bittorrent-portrangel] provides an exhaustive testing report about the behaviour of BitTorrent in an A+P architecture. [draft-boucadair-behave-bittorrent-portrangel] 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 both port range and non-

continuous port sets 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

[RFC6346]

R. Bush., " The Address plus Port (A+P) Approach to the IPv4 Address Shortage", August, 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. Additional Authors

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

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

10. 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.

Appreciation to Randy Bush's initial idea of documenting these experience results, for share the knowledge of what we have learnt with the community.

Thanks to Jan Zorz for comments.

11. Authors' Addresses

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

Email: dxhbupt@gmail.com

Mohamed BOUCADAIR
France Telecom
Rennes, 35000 France

Email: mohamed.boucadair@orange-ftgroup.com

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

Email: Yiu_Lee@Cable.Comcast.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

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 26, 2012

R. Despres
October 24, 2011

Unifying Double Translation and Encapsulation for 4rd (4rd-U)
draft-despres-softwire-4rd-u-01

Abstract

This document proposes a new packet format for IPv4 packets to traverse IPv6 networks. Its purpose is to get, for Residual Deployment of public IPv4 across IPv6 networks (4rd), the best of Encapsulation and Double-translation solutions. For this, it ensures end-to-end transparency of IPv6 networks to IPv4 packets, and makes it possible to configure existing IPv6 Operation & Management tools so that they operate also on 4rd 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 April 26, 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. Problem Statement	3
2. The 4rd-U Header Mapping	6
3. 4rd-U Address Mapping - Checksum Neutrality	8
4. Conclusion	12
5. Acknowledgements	12
6. References	12
6.1. Normative References	12
6.2. Informative References	12
Author's Address	13

1. Problem Statement

Stateless solutions for residual deployment of IPv4 across IPv6 networks, with shared and non-shared IPv4 addresses, have been found desirable by a number of operators (ref [I-D.ietf-software-stateless-4v6-motivation]). The most general configurations to be supported are shown in Figure 1, where CE stands for Customer-edge router, and BR for IPv6-network Border router.

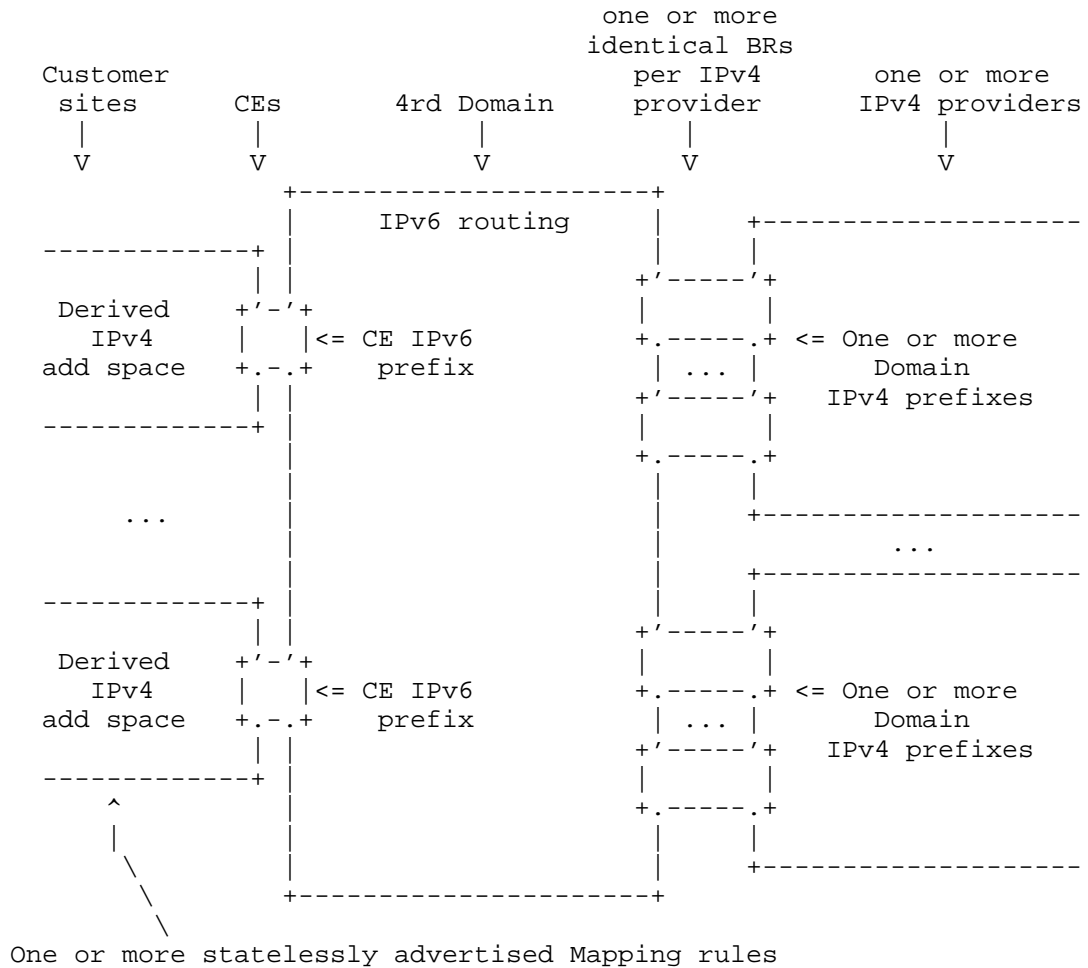


Figure 1: 4rd Domain Model

For IPv6-domain traversal, different IPv6 packet formats have been proposed to convey IPv4 packets. Some are based on Double translation (e.g. in [I-D.xli-behave-divi] and [I-D.xli-behave-divi-pd]); some are based on Encapsulation (e.g. in [I-D.murakami-softwire-4rd]). Both formats having advantages of their own, this document proposes to combine them in a unified approach (4rd-U).

An important advantage of Encapsulation is that it fully preserves network transparency to IPv4 packets, while Double translation, based on the IP/ICMP translation algorithm of [RFC6145], introduces the following limitations to network transparency:

- o IPv4 options at the IP layer are not translated.
- o The "don't fragment" bit of IPv4 (DF bit) is not translated.
- o The IPv4 Type-of-service octet (TOS) cannot be preserved if the traversed IPv6 network has constraints on the IPv6 Traffic-class octet (TC).

The first limitation, lack of IPv4-option support, can be accepted for the following reasons: (1) IPv4 options are very rarely used; (2) they don't influence which applications are supported; (3) Error messages are available to inform sources that options they tried to use are not supported ([RFC0792] has an ICMP error messages meaning "something is wrong with the type code of the first option").

The second limitation, lack of preservation of the DF-bit by IPv4-IPv6 translators, is more problematic:

Its origin is that IPv4 and IPv6 treat packet fragmentations differently. In IPv4, sources MAY fragment packets and indicate, on a per packet basis, whether the network MAY itself fragment packets or not [RFC0791]: if a packet has its DF = 1 and is too big for the next link to be traversed without fragmentation, the network MUST discard this packet (the ICMP error message specified for this case in [RFC0792] is "fragmentation needed and DF set"); if a packet has its DF = 0 and is too big for the next link to be traversed without fragmentation, the network MUST fragment it, and forward its fragments. In IPv6, sources MAY fragment packets, but networks MUST NOT fragment them: if a packet is too big for the next link to be traversed without fragmentation, the network MUST discard it (with the returned ICMPv6 error message "Packet too big" [RFC4443]).

Now, if an IPv4 packet having DF = 0 is too long to traverse an IPv6 network as a single packet (e.g. has 1400 octets and needs to traverse an IPv6 link whose limit is 1280 octets), the IPv4 packet MUST be fragmented and forwarded to comply with IPv4 rules, independently of the value of its DF bit. The problem of translation is then that IPv6 packets have no field to convey the DF-bit value. At IPv6 network exit, where the IPv6 packet has to be translated back to IPv4, one cannot determine whether: (a) the IPv4 packet had been fragmented by its source with DF = 1 and with a size small enough to need no fragmentation to traverse the IPv6 network; or, (b), the IPv4 packet had been sent with DF = 0 and needed fragmentation to traverse the IPv6 network. The original DF bit is lost.

Consequences of not complying end-to-end with the IPv4-fragmentation specification may be limited, but there is no way to guarantee they will remain negligible. If choice exists, solutions that avoid this risk SHOULD therefore be preferred.

The third limitation, lack of guaranteed transparency to the IPv4 TOS, has consequences that are difficult to predict because of the diversity of existing supports of TOS and TC octets. In some IPv6 networks, the IPv4 TOS can without problem be mapped into the IPv6 TC at IPv6-network entrance, and mapped back to the IPv4 TOS at IPv6-network exit. But in some other IPv6 networks, the IPv6 TC to be used for domain traversal has to comply with local constraints. To avoid that these constraints interfere with the semantics of the IPv4 TOS, the original TOS at domain exit MUST in this case be restored.

On the other hand, Double translation has a significant advantage over Encapsulation: a number of existing Operation & Maintenance functions that work for IPv6 can be configured to work also for IPv4, even if concerned with transport-layer ports (e.g. Access control lists), or with valid transport-layer checksums (e.g. for web redirection).

The problem is then to find a new design, if it exists, that keeps the best of both proposed approaches: (a) end-to-end transparency to IPv4; (b) applicability of IPv6 Operation & Maintenance tools of IPv6-only domains to IPv4 packets.

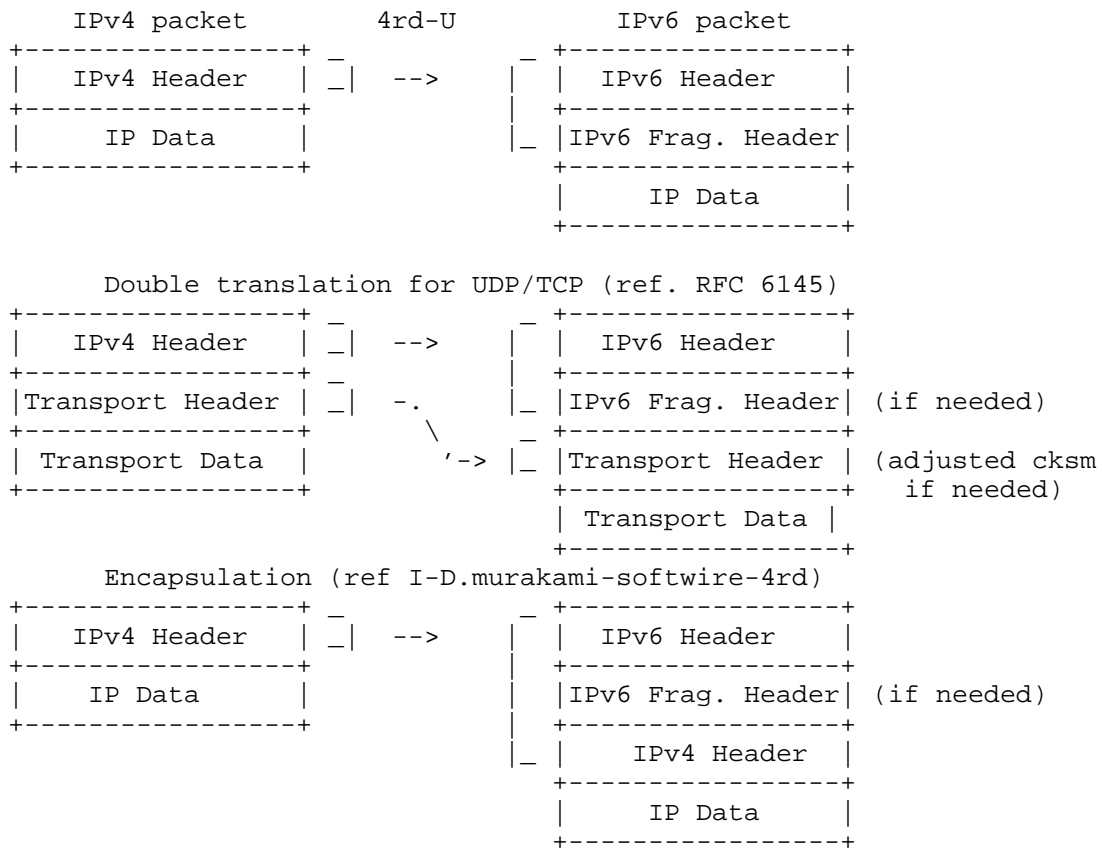
Such a design happening to be possible, it is proposed in Section 2.

2. The 4rd-U Header Mapping

The approach of 4rd-U to meet requirements of Section 1 consists in specifying a header mapping from IPv4 to IPv6 that:

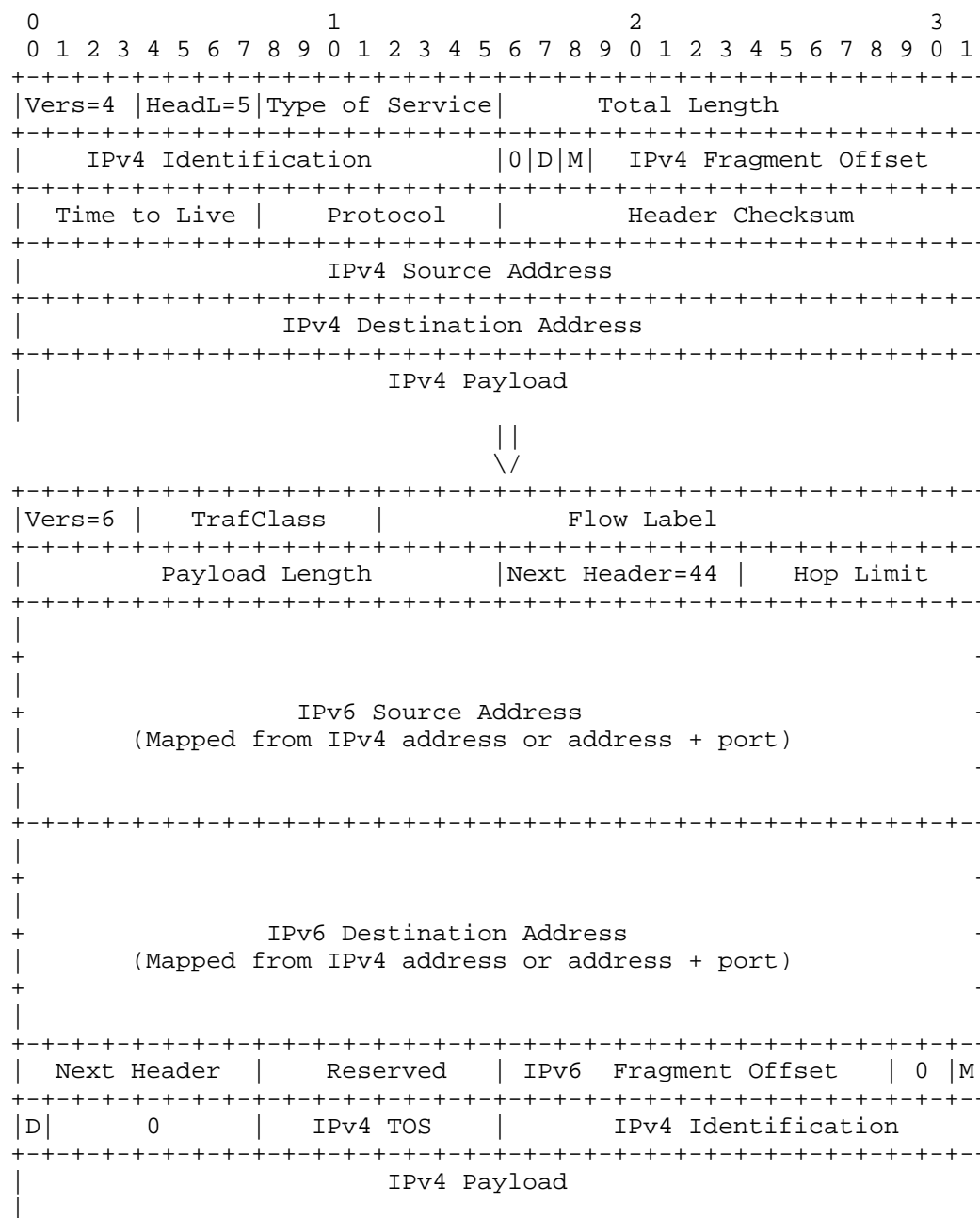
- o is reversible (network transparency to IPv4),
- o has no impact on UDP and TCP checksums (checksum neutrality).

Its IPv6 packet format, compared to those of Double Translation and those of Encapsulation is shown in Figure 2.



Compared Packet Formats of 4rd-U - Translation - Encapsulation

Figure 2



4rd-U Header Mapping

Figure 3

The 4rd-U header mapping is detailed in Figure 3. In it, D stands for DF, and the Flow Label should be set to 0. Values of all other fields can be determined by simple rules known by anyone familiar with IPv4 and IPv6 packet formats.

This header mapping takes advantage of the following favorable technical facts:

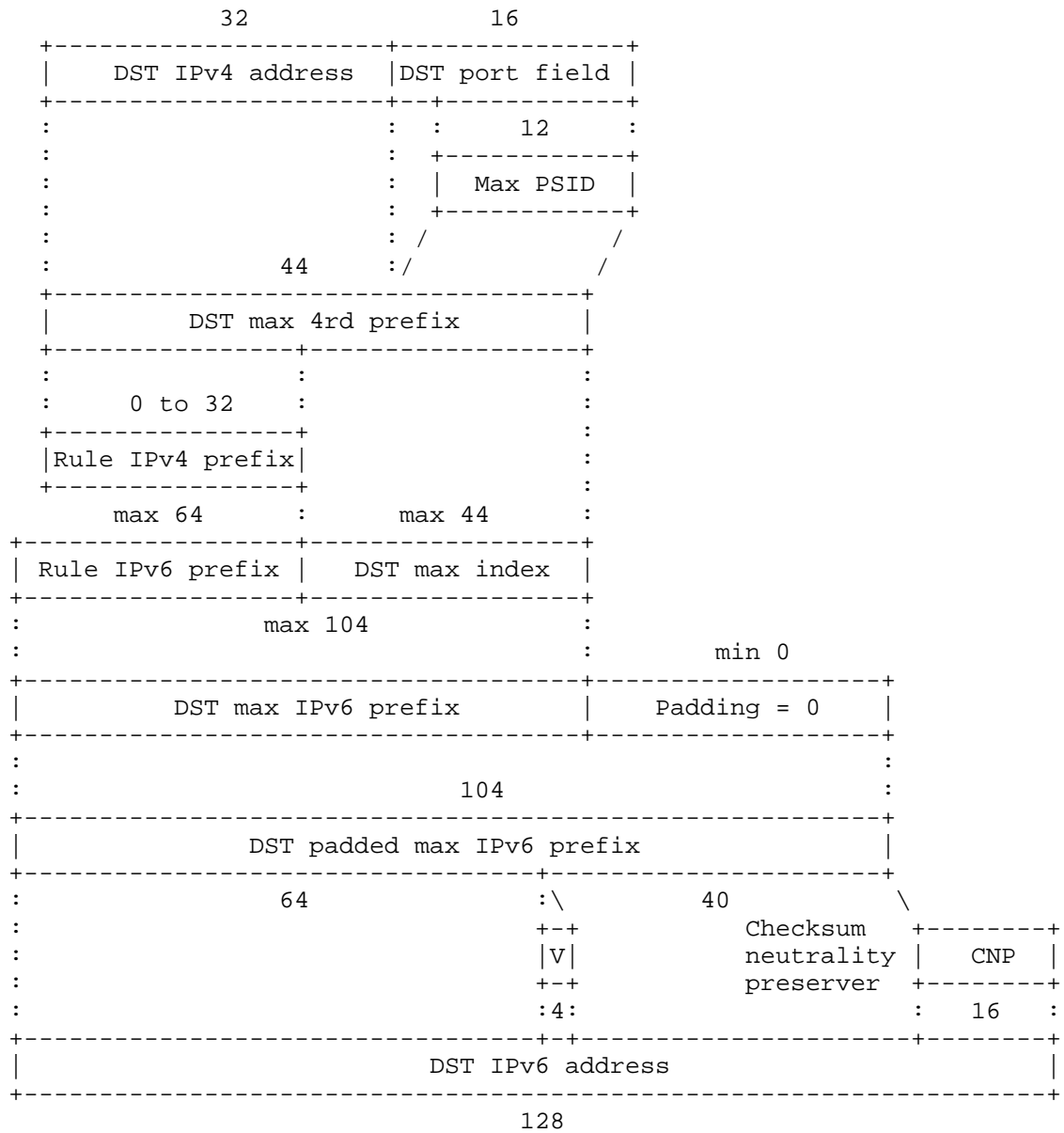
- o IPv6 packets MAY contain Fragment headers
- o In IPv6 fragment headers, Identification fields used for reassembly have 64 bits, i.e. 32 bits more than in IPv4. (This is enough to transparently convey IPv4 fields that need to be restorable at IPv6-domain exit, namely the DF bit and the TOS octet as explained in Section 1).
- o IPv6 addresses used to convey IPv4 packets can be made checksum neutral (see Section 3).

3. 4rd-U Address Mapping - Checksum Neutrality

For residual IPv4 deployments across IPv6 networks, a number of address-mappings between IPv4 addresses, or IPv4 addresses plus ports, and IPv6 addresses have been proposed. One of them has even been proposed as a unified address mapping for Double translation and Encapsulation solutions [unifAddMapp]. But it was devised before a further unification, with a unified packet format like that of 4rd-U, had been envisaged. It misses one important feature: the checksum neutrality discussed in Section 2.

The proposal below, is devised to keep all significant properties of [unifAddMapp] and to add checksum neutrality. It sacrifices for this the length indicator that was included in IPv6 addresses but was necessary neither for routing 4rd packets in IPv6 networks, nor for 4rd functions at exit of IPv6 networks. (It could have been useful to facilitate maintenance, but in a way that appears quite secondary.)

Figure 4 shows steps whereby an IPv6 destination address is derived from an IPv4 destination address and from the value of the field where transport protocols have their destination ports ("DST port field"). If the IP payload has less than 4 octets, missing octets of the DST port field are replaced by 0s to keep uniformity of treatment.



From DST IPv4 Address plus DST Port field to DST IPv6 address

Figure 4

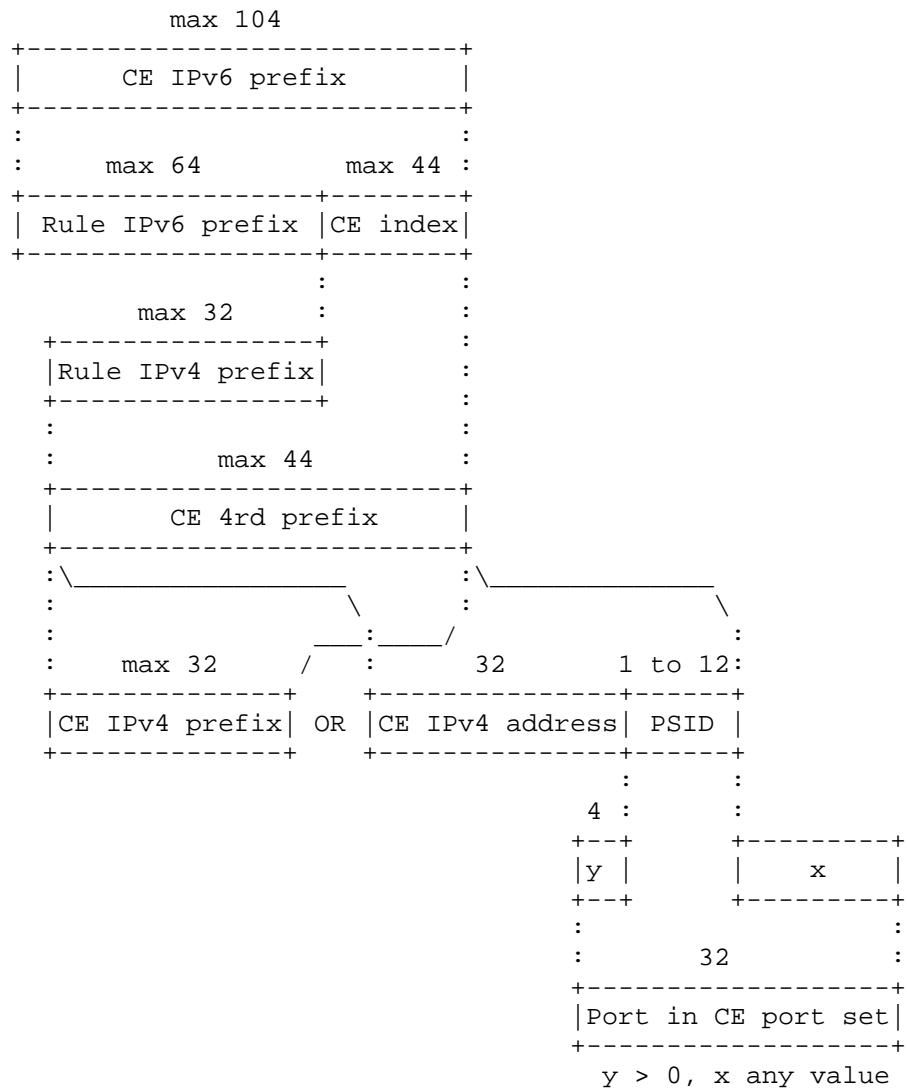
Successive steps are the following:

1. Build the DST max 4rd prefix by concatenating the DST IPv4 address and the last 12 bits of the DST port field.
2. Build the DST padded max IPv6 prefix by replacing, in this DST max 4rd prefix, bits that match the IPv4 prefix of a mapping rule by the IPv6 prefix of this mapping rule, and by padding the result with 0s if necessary to reach 104 bits.
3. Build the DST IPv6 address by concatenating: (a) the first 64 bits of the DST padded max IPv6 prefix; (b) The 4rd V octet, a mark that never appears in any other IPv6 address, and whose proposed value is 0x03 (it permits to use the same CE IPv6 prefix for 4rd packets and for other IPv6 packets); (c) the last 40 bits of the DST padded max IPv6 prefix; (d) A Checksum neutrality preserver CNP (in one's complement arithmetic, it is the sum of the two 16 bit fields of the IPv4 address minus the sum of 16 bit fields of (a).(b).(c)).

Unless something has been missed, this address mapping has, in addition to its checksum neutrality, all properties needed for scenarios of previously documented Encapsulation and Double-translation solutions.

In particular, with a mapping rule whose IPv4 prefix is 0/0, it can support scenarios where full IPv4 addresses are contained in IPv6 addresses, e.g. those of [I-D.xli-behave-divi].

It can also support scenarios where IPv6 routing plans are independent from any consideration on IPv4, like those permitted by [I-D.despres-software-4rd-addmapping]. For these, the way in which CEs derive their IPv4 prefixes, IPv4 addresses, or IPv4 addresses plus restricted port sets, from their IPv6 pre-assigned prefixes, needs no change to be consistent with the 4rd-U address mapping as specified above (see Figure 5).



From IPv6 prefix to IPv4 prefix or IPv4 address + Port set

Figure 5

4. Conclusion

This proposal is submitted to the Softwire WG as a short term technical contribution, not as a document expected to become per se an RFC.

The expectation is that its substance, duly evaluated by the WG, and amended as much as found necessary, can quickly be a basis of a unified standard, suitable for all desirable scenarios for stateless deployments of residual IPv4 across IPv6 networks.

5. Acknowledgements

The original idea of unifying Encapsulation and Double translation has been influenced by thorough discussions, at the Softwire interim meeting in Beijing, on compared merits of these two approaches. In particular, contributions of Xing Li, Congxiao Bao, and Wojciech Dec, have to be acknowledged.

Improvements made since the first version of this document have been influenced by remarks received, on the Softwire mailing list and privately, in particular from Satoru Matsushima, Mark Townsley, and Maoke Chen.

6. References

6.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

6.2. Informative References

- [I-D.despres-softwire-4rd-addmapping]
Despres, R., Qin, J., Perreault, S., and X. Deng,
"Stateless Address Mapping for IPv4 Residual Deployment (4rd)", draft-despres-softwire-4rd-addmapping-01 (work in progress), September 2011.
- [I-D.ietf-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-ietf-softwire-stateless-4v6-motivation-00 (work in progress), September 2011.

[I-D.murakami-softwire-4rd]

Murakami, T., Troan, O., and S. Matsushima, "IPv4 Residual Deployment on IPv6 infrastructure - protocol specification", draft-murakami-softwire-4rd-01 (work in progress), September 2011.

[I-D.xli-behave-divi]

Bao, C., Li, X., Zhai, Y., and W. Shang, "dIVI: Dual-Stateless IPv4/IPv6 Translation", draft-xli-behave-divi-03 (work in progress), July 2011.

[I-D.xli-behave-divi-pd]

Li, X., Bao, C., Dec, W., Asati, R., Xie, C., and Q. Sun, "dIVI-pd: Dual-Stateless IPv4/IPv6 Translation with Prefix Delegation", draft-xli-behave-divi-pd-01 (work in progress), September 2011.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

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

[unifAddMapp]

"Proposed Unified Address Mapping for encapsulation and double-translation - <http://www.ietf.org/mail-archive/web/softwires/current/msg02994.html>", October 2011.

Author's Address

Remi Despres
3 rue du President Wilson
Levallois,
France

Email: despres.remi@laposte.net

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: November 30, 2012

Y. Fu
S. Jiang
Huawei Technologies Co., Ltd
J. Dong
Y.Chen
Tsinghua University
May 28, 2012

DS-Lite Management Information Base (MIB)
draft-fu-softwire-dslite-mib-05

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 November 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.

Abstract

This memo defines a portion of the Management Information Base (MIB) for using with network management protocols in the Internet community. In particular, it defines managed objects for DS-Lite.

Table of Contents

1. Introduction	3
2. The Internet-Standard Management Framework	3
3. Terminology	3
4. Difference from the IP tunnel MIB and NAT MIB	3
5. Relationship to the IF-MIB	5
6. Structure of the MIB Module	5
6.1. The dsliteTunnel Subtree	5
6.2. The dsliteNAT Subtree	5
6.3. The dsliteInfo Subtree	6
6.4. The dsliteTrap Subtree	6
6.5. The dsliteConformance Subtree	6
7. MIB modules required for IMPORTS	6
8. Definitions	6
9. Extending this MIB for Gateway Initiated Dual-Stack Lite.....	27
10. IANA Considerations	27
11. Security Considerations	28
12. References	28
12.1. Normative References	28
12.2. Informative References	29
13. Change Log [RFC Editor please remove]	29
Author's Addresses	30

1. Introduction

Dual-Stack Lite [RFC 6333] is a solution to offer both IPv4 and IPv6 connectivity to customers crossing IPv6 only infrastructure. One of its key components is an IPv4-over-IPv6 tunnel, which is used to provide IPv4 connection across service provider IPv6 network. Another key component is a carrier-grade IPv4-IPv4 NAT to share service provider IPv4 addresses among customers.

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. This MIB module may be used for configuration and monitoring the devices in the Dual-Stack Lite scenario. This MIB also can be extended to the application for Gateway Initiated Dual-Stack Lite.

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 [RFC3410].

Managed objects are accessed via a virtual information store, termed the 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 [RFC2578], [RFC2579] and [RFC2580].

3. 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].

4. Difference from the IP tunnel MIB and NAT MIB

The key technologies for DS-Lite are IP in IP (IPv4-in-IPv6) tunnel and NAT (IPv4 to IPv4 translation).

Notes: According to the section 5.2 of RFC6333, DS-Lite only defines IPv4 in IPv6 tunnels at this moment, but other types of encapsulation could be defined in the future. So this DS-Lite MIB only support IP

in IP encapsulation, if the RFC6333 defined other tunnel types in the future, this DS-Lite MIB will be updated then.

The NAT-MIB [RFC4008] is designed to carry translation from any address family to any address family, therefore supports IPv4 to IPv4 translation.

The tunnel MIB [RFC4087] is designed for managing tunnels of any type over IPv4 and IPv6 networks, therefore supports IP in IP tunnels.

However, NAT MIB and tunnel MIB together are not sufficient to support DS-Lite. This document describes the specific MIB requirements for DS-Lite, as below.

In DS-Lite scenario, the tunnel type is IP in IP, more precisely, is IPv4 in IPv6. Therefore, it is unnecessary to describe tunnel type in DS-Lite MIB.

In DS-Lite scenario, the translation type is IPv4 private address to IPv4 public address. Therefore, it is unnecessary to describe the type of address in the corresponding tunnelIfLocalInetAddress and tunnelIfRemoteInetAddress objects in DS-Lite MIB.

In DS-Lite scenario, the AFTR is not only the tunnel end concentrator, but also a 4-4 translator. Within the AFTR, tunnel information and translation information MUST be mapped each other. Two independent MIB is not able to reflect this mapping relationship. Therefore, a combined MIB is necessary.

If the Gateway Initiated Dual-Stack Lite scenario[I-D.ietf-softwire-gateway-init-ds-lite] is required, the MIB defined in this document could be easily extended for GI-DS-Lite. CID (Context Identifier) can be extended to the tunnel MIB to identifier the access devices which have the same IPv4 address. And both CID and SWID (Softwire Identifier) can be extended to the NAT MIB for performing the NAT binding look up.

The implementation of the IP Tunnel MIB is required for DS-Lite. The tunnelIfEncapsMethod in the tunnelIfEntry should be set to dsLite("xx"), and corresponding entry in the DS-Lite module will exist for every tunnelIfEntry with this tunnelIfEncapsMethod. The tunnelIfRemoteInetAddress must be set to ::.

5. 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. Tunnels are handled by creating a logical interface (ifEntry) for each tunnel. DS-Lite 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 represent whether the DS-Lite tunnel function has been originated. The ifInUcastPkts defined in ifTable will represent the number of IPv6 packets which have been encapsulated with IPv4 packets in it. The ifOutUcastPkts defined in ifTable contains the number of IPv6 packets which can be decapsulated to IPv4 in the virtual interface. Also, the IF-MIB defines ifMtu for the MTU of this tunnel interface, so DS-Lite MIB does not need to define the MTU for tunnel.

6. Structure of the MIB Module

The DS-Lite MIB provides a way to configure and manage the devices (AFTRs) in DS-Lite scenario through SNMP.

DS-Lite MIB is configurable on a per-interface basis. It depends on several parts of the IF-MIB [RFC2863], tunnel MIB [RFC4087], and NAT MIB [RFC4008].

6.1. The dsliteTunnel Subtree

The dsliteTunnel subtree describes managed objects used for managing tunnels in the DS-Lite scenario. Because some objects defined in Tunnel MIB are not access, a few new objects are defined in DS-Lite MIB.

6.2. The dsliteNAT Subtree

The dsliteNAT Subtree describes managed objects used for configuration as well as monitoring of AFTR which is capable of NAT function. Because the NAT MIB supports the NAT management function in DS-Lite, we may reuse it in DS-Lite MIB. The dsliteNAT Subtree also provides the information of mapping relationship between the tunnel MIB and NAT MIB by extending B4 address to the bind table in NAT MIB.

6.3. The dsliteInfo Subtree

The dsliteInfo Subtree provides the statistical information for DS-lite.

6.4. The dsliteTrap Subtree

The dsliteTrap Subtree provides trap information in DS-lite instance.

6.5. The dsliteConformance Subtree

The Subtree provides conformance information of MIB objects.

7. MIB modules required for IMPORTS

This MIB module IMPORTs objects from [RFC4008], [RFC2580], [RFC2578], [RFC2863], [RFC4001], [RFC3411].

8. Definitions

```
DSLite-MIB DEFFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE, mib-2, transmission,
    Gauge32, Integer32, Counter64
    FROM SNMPv2-SMI
```

```
    RowStatus, StorageType, DisplayString
    FROM SNMPv2-TC
```

```
    ifIndex, InterfaceIndexOrZero
    FROM IF-MIB
```

```
    IANA tunnelType
    FROM IANAifType-MIB
```

```
    InetAddress, InetAddressIPv6, InetPortNumber
    FROM INET-ADDRESS-MIB
```

```
    NatAddrMapId, natAddrMapName, natAddrMapEntryType,
    natAddrMapLocalAddrFrom, natAddrMapLocalAddrTo,
    natAddrMapLocalPortFrom, natAddrMapLocalPortTo,
    natAddrMapGlobalAddrFrom, natAddrMapGlobalAddrTo,
    natAddrMapGlobalPortFrom, natAddrMapGlobalPortTo,
    natAddrPortBindGlobalAddr, natAddrPortBindGlobalPort,
    NatBindId, natAddrPortBindSessions,
    natAddrPortBindMaxIdleTime, natAddrPortBindCurrentIdleTime,
```

natAddrPortBindInTranslates, natAddrPortBindOutTranslates
FROM natMIB

dsliteMIB MODULE-IDENTITY

LAST-UPDATED "201205280000Z" -- May 28, 2012

ORGANIZATION "IETF Softwire Working Group"

CONTACT-INFO

"Yu Fu

Huawei Technologies Co., Ltd
Huawei Building, 156 Beiqing Rd., Hai-Dian District
Beijing, P.R. China 100095
EMail: eleven.fuyu@huawei.com

Sheng Jiang

Huawei Technologies Co., Ltd
Huawei Building, 156 Beiqing Rd., Hai-Dian District
Beijing, P.R. China 100095
EMail: jiangsheng@huawei.com

Jiang Dong

Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R. China
Email: dongjiang@csnet1.cs.tsinghua.edu.cn

Yuchi Chen

Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R. China
Email: flashfoxmx@gmail.com "

DESCRIPTION

"The MIB module is defined for management of object in the
DS-Lite scenario. "

::= { transmission xxx } --xxx to be replaced with correct
value

dsliteTunnel OBJECT IDENTIFIER

:: = { dsliteMIB 1 }

dsliteNAT OBJECT IDENTIFIER

:: = { dsliteMIB 2 }

dsliteInfo OBJECT IDENTIFIER

:: = { dsliteMIB 3 }

```

dsliteTraps OBJECT IDENTIFIER
 ::= { dsliteMIB 4 }

--Conformance
dsliteConformance OBJECT IDENTIFIER
 ::= { dsliteMIB 5 }

--dsliteTunnel
--dsliteTunnelTable

dsliteTunnelTable OBJECT-TYPE
 SYNTAX      SEQUENCE OF dsliteTunnelEntry
 MAX-ACCESS  not-accessible
 STATUS      current
 DESCRIPTION
  "The (conceptual) table containing information on configured
   tunnels. This table can be used to map CPE address to the
   associated AFTR address. It can also be used for row
   creation."
 ::= { dsliteTunnel 1 }

dsliteTunnelEntry OBJECT-TYPE
 SYNTAX      dsliteTunnelEntry
 MAX-ACCESS  not-accessible
 STATUS      current
 DESCRIPTION
  "Each entry in this table contains the information on a
   particular configured tunnel."
  INDEX      { dsliteTunnelStartAddress,
               dsliteTunnelEndAddress,
               ifIndex }
 ::= { dsliteTunnelTable 1 }

dsliteTunnelEntry ::=
 SEQUENCE {
  dsliteTunnelStartAddress      InetAddressIPv6,
  dsliteTunnelStartAddPreLen    Integer32,
  dsliteTunnelEndAddress        InetAddressIPv6,
  dsliteTunnelStatus            RowStatus,
  dsliteTunnelStorageType       StorageType
 }

dsliteTunnelStartAddress OBJECT-TYPE
 SYNTAX      InetAddressIPv6
 MAX-ACCESS  read-create
 STATUS      current
 DESCRIPTION

```



```
        "The address of the start point of the tunnel."
 ::= { dsliteTunnelEntry 1 }

dsliteTunnelStartAddPreLen OBJECT-TYPE
    SYNTAX Integer32 (0..128)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "IPv6 prefix length of the IP address of the
         start point of the tunnel."
 ::= { dsliteTunnelEntry 2 }

dsliteTunnelEndAddress OBJECT-TYPE
    SYNTAX      InetAddressIPv6
    MAX-ACCESS read-create
    STATUS      current
    DESCRIPTION
        "The address of the endpoint of the tunnel."
 ::= { dsliteTunnelEntry 3 }

dsliteTunnelStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS read-create
    STATUS      current
    DESCRIPTION
        "The status of this row, by which new entries may be
         created, or old entries deleted from this table."
 ::= { dsliteTunnelEntry 4 }

dsliteTunnelStorageType OBJECT-TYPE
    SYNTAX      StorageType
    MAX-ACCESS read-create
    STATUS      current
    DESCRIPTION
        "The storage type of this row. If the row is
         permanent(4), no objects in the row need be
         writable."
 ::= { dsliteTunnelEntry 5 }

--dsliteNAT
--dsliteNATMapTable(define address pool)
--dsliteNATBindTable

dsliteNATMapTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF dsliteNATMapEntry
    MAX-ACCESS not-accessible
```

```

STATUS      current
DESCRIPTION
    "This table contains information about address map
    parameters."
:: = { dsliteNAT 1 }

dsliteNATMapEntry OBJECT-TYPE
SYNTAX      dsliteNATMapEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    " This entry represents an address map to be used for
    NAT and contributes to the address mapping tables of
    AFTR."
INDEX       { ifIndex,
              dsliteNATMapIndex }
:: = { dsliteNATMapTable 1 }

dsliteNATMapEntry ::=
SEQUENCE {
dsliteNATMapIndex          NatAddrMapId,
dsliteNATMapAddrName       natAddrMapName,
dsliteNATMapEntryType      natAddrMapEntryType,
dsliteNATMapLocalAddrFrom  natAddrMapLocalAddrFrom,
dsliteNATMapLocalAddrTo    natAddrMapLocalAddrTo,
dsliteNATMapLocalPortFrom  natAddrMapLocalPortFrom,
dsliteNATMapLocalPortTo    natAddrMapLocalPortTo,
dsliteNATMapGlobalAddrFrom natAddrMapGlobalAddrFrom,
dsliteNATMapGlobalAddrTo   natAddrMapGlobalAddrTo,
dsliteNATMapGlobalPortFrom natAddrMapGlobalPortFrom,
dsliteNATMapGlobalPortTo   natAddrMapGlobalPortTo,
dsliteNATMapAddrUsed       natAddrMapAddrUsed,
dsliteNATMapStorageType    StorageType,
dsliteNATMapRowStatus      RowStatus
}

dsliteNATMapIndex OBJECT-TYPE
SYNTAX      NatAddrMapId
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "Along with ifIndex, this object uniquely
    identifies an entry in the dsliteNATMapTable.
    Address map entries are applied in the order
    specified by dsliteNATMapIndex."
::= { dsliteNATMapEntry 1 }

```

```
dsliteNATMapAddrName OBJECT-TYPE
    SYNTAX      natAddrMapName
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "Name identifying all map entries in the table associated
        with the same interface. All map entries with the same
        ifIndex MUST have the same map name."
    ::= { dsliteNATMapEntry 2 }

dsliteNATMapEntryType OBJECT-TYPE
    SYNTAX      natAddrMapEntryType
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "This parameter can be used to set up static
        or dynamic address maps."
    ::= { dsliteNATMapEntry 3 }

dsliteNATMapLocalAddrFrom OBJECT-TYPE
    SYNTAX      natAddrMapLocalAddrFrom
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "This object specifies the first IP address of the range
        of IP addresses mapped by this translation entry.
        The value of this object must be less than or
        equal to the value of the dsliteNATMapLocalAddrTo
        object."
    ::= { dsliteNATMapEntry 4 }

dsliteNATMapLocalAddrTo OBJECT-TYPE
    SYNTAX      natAddrMapLocalAddrTo
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "This object specifies the last IP address of the range of
        IP addresses mapped by this translation entry. If only
        a single address is being mapped, the value of this
        object is equal to the value of natAddrMapLocalAddrFrom.
        The value of this object must be greater than or equal to
        the value of the natAddrMapLocalAddrFrom object."
    ::= { dsliteNATMapEntry 5 }

dsliteNATMapLocalPortFrom OBJECT-TYPE
    SYNTAX      natAddrMapLocalPortFrom
    MAX-ACCESS   read-create
```

```
STATUS      current
DESCRIPTION
    "The value of this object must be less than or equal
    to the value of the dsliteNATMapLocalPortTo object.
    If the translation specifies a single port, then the
    value of this object is equal to the value of
    dsliteNATMapLocalPortTo."
DEFVAL { 0 }
 ::= { dsliteNATMapEntry 6 }

dsliteNATMapLocalPortTo OBJECT-TYPE
SYNTAX      natAddrMapLocalPortTo
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The value of this object must be greater than or equal
    to the value of the dsliteNATMapLocalPortFrom object.
    If the translation specifies a single port, then
    the value of this object is equal to the value of
    dsliteNATMapLocalPortFrom."
DEFVAL { 0 }
 ::= { dsliteNATMapEntry 7 }

dsliteNATMapGlobalAddrFrom OBJECT-TYPE
SYNTAX      natAddrMapGlobalAddrFrom
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "This object specifies the first IP address of
    the range of IP addresses being mapped to.
    The value of this object must be less than
    or equal to the value of the
    dsliteNATMapGlobalAddrTo object.
 ::= { dsliteNATMapEntry 8 }

dsliteNATMapGlobalAddrTo OBJECT-TYPE
SYNTAX      natAddrMapGlobalAddrTo
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "This object specifies the last IP address of the range
    of IP addresses being mapped to. If only a single
    address is being mapped to, the value of this object
    is equal to the value of dsliteNATMapGlobalAddrFrom.
    The value of this object must be greater than or equal
    to the value of the dsliteNATMapGlobalAddrFrom object.
 ::= { dsliteNATMapEntry 9 }
```

```
dsliteNATMapGlobalPortFrom OBJECT-TYPE
    SYNTAX      natAddrMapGlobalPortFrom
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The value of this object must be less than or equal
        to the value of the dsliteNATMapGlobalPortTo object.
        If the translation specifies a single port, then the
        value of this object is equal to the value
        dsliteNATMapGlobalPortTo."
    DEFVAL { 0 }
    ::= { dsliteNATMapEntry 10 }

dsliteNATMapGlobalPortTo OBJECT-TYPE
    SYNTAX      natAddrMapGlobalPortTo
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The value of this object must be greater than or
        equal to the value of the dsliteNATMapGlobalPortFrom
        object. If the translation specifies a single port,
        then the value of this object is equal to the
        value of dsliteNATMapGlobalPortFrom."
    DEFVAL { 0 }
    ::= { dsliteNATMapEntry 11 }

dsliteNATMapAddrUsed OBJECT-TYPE
    SYNTAX      natAddrMapAddrUsed
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of addresses pertaining to this address
        map that are currently being used from the NAT pool."
    ::= { dsliteNATMapEntry 12 }

dsliteNATMapStorageType OBJECT-TYPE
    SYNTAX      StorageType
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The storage type for this conceptual row.
        Conceptual rows having the value 'permanent'
        need not allow write-access to any columnar
        objects in the row."
    REFERENCE
        "Textual Conventions for SMIV2, Section 2."
```

```

    DEFVAL { nonVolatile }
    ::= { dsliteNATMapEntry 13 }

dsliteNATMapRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The status of this conceptual row."
    REFERENCE
        "Textual Conventions for SMIV2, Section 2."
    ::= { dsliteNATMapEntry 14 }

dsliteNATBindTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF dsliteNATBindEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table contains information about currently
        active NAT binds in AFTR. This table extends the
        natAddrPortBindTable designed in NAT MIB (RFC
        4008) by IPv6 address of B4."
    ::= { dsliteNAT 2 }

dsliteNATBindEntry OBJECT-TYPE
    SYNTAX      dsliteNATBindEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "Each entry in this table holds the relationship between
        tunnel information and nat bind information. These entries
        are lost upon agent restart."
    INDEX       { ifIndex,
                  dsliteNATBindLocalAddr,
                  dsliteNATBindLocalPort,
                  dsliteB4Addr }
    ::= { dsliteNATBindTable 1 }

dsliteNATBindEntry ::= =
    SEQUENCE {
        dsliteNATBindLocalAddr      InetAddress,
        dsliteNATBindLocalPort      InetPortNumber,
        dsliteNATBindGlobalAddr     natAddrPortBindGlobalAddr,
        dsliteNATBindGlobalPort     natAddrPortBindGlobalPort,
        dsliteNATBindId             NatBindId,
        dsliteB4Addr                dsliteTunnelStartAddress,
        dsliteB4PreLen              dsliteTunnelStartAddPreLen,

```

```
    dsliteNATBindMapIndex          NatAddrMapId,
    dsliteNATBindSessions          natAddrPortBindSessions,
    dsliteNATBindMaxIdleTime       natAddrPortBindMaxIdleTime,
    dsliteNATBindCurrentIdleTime   natAddrPortBindCurrentIdleTime,
    dsliteNATBindInTranslates      natAddrPortBindInTranslates,
    dsliteNATBindOutTranslates     natAddrPortBindOutTranslates
}

dsliteNATBindLocalAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object represents the private IP address of host."
    ::= { dsliteNATBindEntry 1 }

dsliteNATBindLocalPort OBJECT-TYPE
    SYNTAX      InetPortNumber
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object represents the private-realm Port
        number of host."
    ::= { dsliteNATBindEntry 2 }

dsliteNATBindGlobalAddr OBJECT-TYPE
    SYNTAX      natAddrPortBindGlobalAddr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object represents the public-realm IP
        address of host."
    ::= { dsliteNATBindEntry 3 }

dsliteNATBindGlobalPort OBJECT-TYPE
    SYNTAX      natAddrPortBindGlobalPort
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object represents the public-realm Port number
        of host."
    ::= { dsliteNATBindEntry 4 }

dsliteNATBindId OBJECT-TYPE
    SYNTAX      NatBindId
    MAX-ACCESS  read-only
    STATUS      current
```

DESCRIPTION
"This object represents a bind id that is dynamically assigned to each bind by AFTR. Each bind is represented by a unique bind id across the dsliteNATBindTable."
 ::= { dsliteNATBindEntry 5 }

dsliteB4Addr OBJECT-TYPE
SYNTAX dsliteTunnelStartAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object represents the relationship between tunnel start point to the Bind entry, which extends the source IPv6 address of packet to the Bind table."
 ::= { dsliteNATBindEntry 6 }

dsliteB4PreLen OBJECT-TYPE
SYNTAX dsliteTunnelStartAddPreLen
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object indicates the IPv6 prefix length of the start point of tunnel, which is also need to extend to the Bind table."
 ::= { dsliteNATBindEntry 7 }

dsliteNATBindMapIndex OBJECT-TYPE
SYNTAX NatAddrMapId
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object is a pointer to the dsliteNATMapTable entry used in creating this BIND."
 ::= { dsliteNATBindEntry 8 }

dsliteNATBindSessions OBJECT-TYPE
SYNTAX natAddrPortBindSessions
MAX-ACCESS read-only
STATUS current
DESCRIPTION
" This object represents the number of sessions currently using this BIND."
 ::= { dsliteNATBindEntry 9 }

dsliteNATBindMaxIdleTime OBJECT-TYPE
SYNTAX natAddrPortBindMaxIdleTime


```
    MAX-ACCESS read-only
    STATUS      current
DESCRIPTION
    "This object indicates the maximum time for
    which this bind can be idle without any sessions
    attached to it."
 ::= { dsliteNATBindEntry 10 }

dsliteNATBindCurrentIdleTime OBJECT-TYPE
    SYNTAX      natAddrPortBindCurrentIdleTime
    MAX-ACCESS read-only
    STATUS      current
DESCRIPTION
    "At any given instance, this object indicates the
    time that this bind has been idle without any sessions
    attached to it."
 ::= { dsliteNATBindEntry 11 }

dsliteNATBindInTranslates OBJECT-TYPE
    SYNTAX      natAddrPortBindInTranslates
    MAX-ACCESS read-only
    STATUS      current
DESCRIPTION
    "The number of inbound packets that were
    translated as per this bind entry."
 ::= { dsliteNATBindEntry 12 }

dsliteNATBindOutTranslates OBJECT-TYPE
    SYNTAX      natAddrPortBindOutTranslates
    MAX-ACCESS read-only
    STATUS      current
DESCRIPTION
    "The number of outbound packets that were
    translated as per this bind entry."
 ::= { dsliteNATBindEntry 13 }

--dsliteInfo

dsliteSessionLimitTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF dsliteSessionLimitEntry
    MAX-ACCESS not-accessible
    STATUS      current
DESCRIPTION
    "The (conceptual) table containing information about session
    limit. It can also be used for row creation."
 ::= { dsliteInfo 1 }
```

```
dsliteSessionLimitEntry OBJECT-TYPE
    SYNTAX      dsliteSessionLimitEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each entry in this table contains the information to be
        used for configuring session limits for DS-lite."
    INDEX       { dsliteInstanceName,
                  dsliteSessionLimitaType }
    ::= { dsliteSessionLimitTable 1 }

dsliteSessionLimitEntry ::=
    SEQUENCE {
        dsliteSessionLimitInstanceName      DisplayString,
        dsliteSessionLimitType              INTEGER,
        dsliteSessionLimitNumber            Integer32,
        dsliteSessionLimitStorageType       StorageType,
        dsliteSessionLimitRowStatus         RowStatus
    }

dsliteSessionLimitInstanceName OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (1..31))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        " This object represents the instance name
        that is limited."
    ::= { dsliteSessionLimitEntry 1 }

dsliteSessionLimitType OBJECT-TYPE
    SYNTAX      INTEGER
    {
        tcp(0),
        udp(1),
        icmp(2),
        total(3)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object represents the session limit type :
        tcp or udp or totally."
    ::= { dsliteSessionLimitEntry 2 }

dsliteSessionLimitNumber OBJECT-TYPE
    SYNTAX      Integer32 (1..65535)
    MAX-ACCESS  read-create
```

```
STATUS current
DESCRIPTION
    " This table represents the limit number of the session."
 ::= { dsliteSessionLimitEntry 3 }

dsliteSessionLimitStorageType OBJECT-TYPE
    SYNTAX StorageType
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "The storage type for this conceptual row. Conceptual
        rows having the value 'permanent' need not allow
        write-access to any columnar objects in the row."
    ::= { dsliteSessionLimitEntry 4 }

dsliteSessionLimitRowStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        " The status of this conceptual row."
    REFERENCE
        "Textual Conventions for SMIV2, Section 2."
    DEFVAL { nonVolatile }
    ::= { dsliteSessionLimitEntry 5 }

dslitePortLimitTable OBJECT-TYPE
    SYNTAX SEQUENCE OF dslitePortLimitEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table is used to configure port limits for a
        DS-Lite instance."
    ::= { dsliteInfo 2 }

dslitePortLimitEntry OBJECT-TYPE
    SYNTAX dslitePortLimitEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Each entry in this table contains the information to be
        used for configuring port limits for DS-lite."
    INDEX { dslitePortLimitInstanceName,
            dslitePortLimitType }
    ::= { dslitePortLimitTable 1 }
```

```
dslitePortLimitEntry ::=
    SEQUENCE {
        dslitePortLimitInstanceName      DisplayString,
        dslitePortLimitType               INTEGER,
        dslitePortLimitNumber             Integer32,
        dslitePortLimitStorageType        StorageType,
        dslitePortLimitRowStatus          RowStatus
    }
```

```
dslitePortLimitInstanceName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (1..31))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        " This object represents the instance name
          that is limited."
    ::= { dslitePortLimitEntry 1 }
```

```
dslitePortLimitType OBJECT-TYPE
    SYNTAX INTEGER
    {
        tcp(0),
        udp(1),
        icmp(2),
        total(3)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This object represents the port limit
         type: tcp or udp or totally."
    ::= { dslitePortLimitEntry 2 }
```

```
dslitePortLimitNumber OBJECT-TYPE
    SYNTAX Integer32 (1..300000)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This object represents the limit number of the
         port usage."
    ::= { dslitePortLimitEntry 3 }
```

```
dslitePortLimitStorageType OBJECT-TYPE
    SYNTAX StorageType
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
```

```

        "The storage type for this conceptual row. Conceptual
        rows having the value 'permanent' need not allow
        write-access to any columnar objects in the row."
 ::= { dslitePortLimitEntry 4 }

dslitePortLimitRowStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "Create or delete table row."
    ::= { dslitePortLimitEntry 5 }

dsliteAFTRAlarmScalar OBJECT IDENTIFIER ::= { dsliteInfo 3 }

dsliteAFTRAlarmB4Addr OBJECT-TYPE
    SYNTAX dsliteTunnelStartAddress
    MAX-ACCESS accessible-for-notify
    STATUS current
    DESCRIPTION
        "This object indicate the IP address of
        B4 that send alarm "
    ::= { dsliteAFTRAlarmScalar 1 }

dsliteAFTRAlarmProtocolType OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS accessible-for-notify
    STATUS current
    DESCRIPTION
        "This object indicate the procotol type of alarm,
        0:tcp,1:udp,2:icmp,3:total "
    ::= { dsliteAFTRAlarmScalar 2 }

dsliteAFTRAlarmMapAddrName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS accessible-for-notify
    STATUS current
    DESCRIPTION
        "This object indicate the name of dsliteNATMapAddrName "
    ::= { dsliteAFTRAlarmScalar 3 }

dsliteAFTRAlarmSpecificIP OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS accessible-for-notify
    STATUS current
    DESCRIPTION
        " This object indicate the IP address whose port usage
```

```

        reach threshold "
 ::= { dsliteAFTRAlarmScalar 4 }

dsliteAFTRAlarmConnectNumber OBJECT-TYPE
    SYNTAX Integer32 (60..90)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        " This object indicate the threshold of DS-Lite
          connections alarm."
 ::= { dsliteAFTRAlarmScalar 5 }

dsliteStatisticTable OBJECT-TYPE
    SYNTAX SEQUENCE OF dsliteStatisticEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table provides statistical information
          of DS-Lite."
 ::= { dsliteInfo 4 }

dsliteStatisticEntry OBJECT-TYPE
    SYNTAX dsliteStatisticEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table provides statistical information
          of DS-Lite."
    INDEX { dsliteStatisticInstanceName }
 ::= { dsliteStatisticTable 1 }

dsliteStatisticEntry ::=
    SEQUENCE {
        dsliteStatisticInstanceName          DisplayString,
        dsliteStatisticDiscard                Counter64,
        dsliteStatisticReceived               Counter64,
        dsliteStatisticTransmitted            Counter64,
        dsliteStatisticIpv4Session            Counter64,
        dsliteStatisticIpv6Session            Counter64,
        dsliteStatisticStorageType            StorageType,
        dsliteStatisticRowStatus              RowStatus
    }

dsliteStatisticInstanceName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (1..31))
    MAX-ACCESS read-only
    STATUS current

```

DESCRIPTION

" This object indicate the instance name
that is limited."

::= { dsliteStatisticEntry 1 }

dsliteStatisticDiscard OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-create

STATUS current

DESCRIPTION

" This object indicate the count number of
the discarded packet."

::= { dsliteStatisticEntry 2 }

dsliteStatisticReceived OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object indicate the count number of
received packet count."

::= { dsliteStatisticEntry 3 }

dsliteStatisticTransmitted OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object indicate the count number of
transmitted packet count."

::= { dsliteStatisticEntry 4 }

dsliteStatisticIpv4Session OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-create

STATUS current

DESCRIPTION

" This object indicate the number of the
current IPv4 Session."

::= { dsliteStatisticEntry 5 }

dsliteStatisticIpv6Session OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-create

STATUS current

DESCRIPTION

```

    " This object indicate the number of the
      current IPv6 Session."
 ::= { dsliteStatisticEntry 6 }

dsliteStatisticRowStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "Create or delete table row."
    ::= { dsliteStatisticEntry 7 }

---dslite trap

dsliteTunnelNumAlarm NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION
        "This trap is triggered when dslite tunnel
        reach the threshold."
    ::= { dsliteTraps 1 }

dsliteAFTRUserSessionNumAlarm NOTIFICATION-TYPE
    OBJECTS { dsliteAFTRAlarmProtocolType,
              dsliteAFTRAlarmB4Addr }
    STATUS current
    DESCRIPTION
        " This trap is triggered when sessions of
        user reach the threshold."
    ::= { dsliteTraps 2 }

dsliteAFTRPortUsageOfSpecificIpAlarm NOTIFICATION-TYPE
    OBJECTS { dsliteAFTRAlarmMapAddrName,
              dsliteAFTRAlarmSpecificIP }
    STATUS current
    DESCRIPTION
        "This trap is triggered when used NAT
        ports of map address reach the threshold."
 ::= { dsliteTraps 3 }

--Module Conformance statement

dsliteCompliances OBJECT IDENTIFIER ::= { dsliteConformance 1 }

dsliteCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "Description."
```



```
MODULE -- this module
    MANDATORY-GROUPS { dsliteNATMapGroup,
                        dsliteTunnelGroup }
::= { dsliteCompliances 1 }

dsliteGroups OBJECT IDENTIFIER ::= { dsliteConformance 2 }

dsliteAFTRAlarmScalarGroup OBJECT-GROUP
    OBJECTS { dsliteAFTRAlarmB4Addr, dsliteAFTRAlarmProtocolType,
              dsliteAFTRAlarmMapAddrName, dsliteAFTRAlarmSpecificIP,
              dsliteAFTRAlarmConnectNumber }
    STATUS current
    DESCRIPTION
        " The collection of this objects are used to give the
          information about AFTR alarming Scalar."
::= { dsliteGroups 1 }

dsliteNATMapGroup OBJECT-GROUP
    OBJECTS { dsliteNATMapIndex, dsliteNATMapAddrName,
              dsliteNATMapEntryType, dsliteNATMapLocalAddrFrom,
              dsliteNATMapLocalAddrTo, dsliteNATMapLocalPortFrom,
              dsliteNATMapLocalPortTo, dsliteNATMapGlobalAddrFrom,
              dsliteNATMapGlobalAddrTo, dsliteNATMapGlobalPortFrom,
              dsliteNATMapGlobalPortTo, dsliteNATMapAddrUsed,
              dsliteNATMapStorageType, dsliteNATMapRowStatu }
    STATUS current
    DESCRIPTION
        " The collection of this objects are used to give the
          information about NAT address mapping."
::= { dsliteGroups 2 }

dsliteTunnelGroup OBJECT-GROUP
    OBJECTS { dsliteTunnelStartAddress, dsliteTunnelStartAddPreLen,
              dsliteTunnelEndAddress,
              dsliteTunnelStatus,
              dsliteTunnelStorageType }
    STATUS current
    DESCRIPTION
        " The collection of this objects are used to give the
          information of tunnel in ds-lite."
::= { dsliteGroups 3 }

dsliteNATBindGroup OBJECT-GROUP
    OBJECTS { dsliteNATBindLocalAddr, dsliteNATBindLocalPort,
              dsliteNATBindGlobalAddr, dsliteNATBindGlobalPort,
              dsliteNATBindId, dsliteB4Addr, dsliteB4PreLen,
              dsliteNATBindMapIndex, dsliteNATBindSessions,
```

```
        dsliteNATBindMaxIdleTime,
        dsliteNATBindCurrentIdleTime,
        dsliteNATBindInTranslates,
        dsliteNATBindOutTranslates }
STATUS current
DESCRIPTION
    " The collection of this objects are used to give the
      information about NAT Bind."
 ::= { dsliteGroups 4 }

dsliteSessionLimitGroup OBJECT-GROUP
OBJECTS { dsliteSessionLimitInstanceName,
          dsliteSessionLimitType, dsliteSessionLimitNumber,
          dsliteSessionLimitStorageType,
          dsliteSessionLimitRowStatus }
STATUS current
DESCRIPTION
    " The collection of this objects are used to give the
      information about port limit."
 ::= { dsliteGroups 5 }

dslitePortLimitGroup OBJECT-GROUP
OBJECTS { dslitePortLimitInstanceName,
          dslitePortLimitType, dslitePortLimitNumber,
          dslitePortLimitStorageType,
          dslitePortLimitRowStatus }
STATUS current
DESCRIPTION
    " The collection of this objects are used to give the
      information about port limit."
 ::= { dsliteGroups 6 }

dsliteStatisticGroup OBJECT-GROUP
OBJECTS { dsliteStatisticInstanceName,
          dsliteStatisticDiscard,
          dsliteStatisticReceived,
          dsliteStatisticTransmitted,
          dsliteStatisticIpv4Session,
          dsliteStatisticIpv6Session,
          dsliteStatisticStorageType,
          dsliteStatisticRowStatus }
STATUS current
DESCRIPTION
    " The collection of this objects are used to give the
      statistical information of ds-lite."
 ::= { dsliteGroups 7 }
```

```

dsliteTrapsGroup NOTIFICATION-GROUP
    NOTIFICATIONS { dsliteTunnelNumAlarm,
                    dsliteAFTRUserSessionNumAlarm,
                    dsliteAFTRPortUsageOfSpecificIpAlarm }
    STATUS current
    DESCRIPTION
        "The collection of this objects are used to give the
        trap information of ds-lite."
    ::= { dsliteGroups 8 }

    END

```

9. Extending this MIB for Gateway Initiated Dual-Stack Lite

Similar to DS-lite, GI-DS-lite enables the service provider to share public IPv4 addresses among different customers by combining tunneling and NAT. 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 host the tunneled packets belong to. The MIB defined in this document can easily be extended to use for GI-DS-Lite scenario. New object as CID SHOULD be extended to the dsliteTunnelTable. And a new object as dsliteTunnelID can be defined in DS-Lite MIB as SWID in GI-DS-Lite. Both CID and SWID SHOULD be extended to the dsliteNATBindTable. It will use the combination of CID and SWID as the unique identifier for the end host and store it in the NAT binding entry.

10. 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
-----	-----
DSLite-MIB	{ transmission XXX }

IANAtunnelType ::= TEXTUAL-CONVENTION

```

SYNTAX      INTEGER {
                                dsLite ("XX")      -- dslite tunnel
                                }

```

Notes: As the Appendix A of the IP Tunnel MIB[RFC4087] described that it has already assigned the value direct(2) to indicate the tunnel type is IP in ip tunnel, but it is still difficult to distinguish the DS-Lite tunnel packets and the normal IP in IP tunnel packets in the scenario of the AFTR connecting to both the DS-lite tunnel and IP in IP tunnel.

11. Security Considerations

The DS-Lite MIB module can be used for configuration of certain objects, and anything that can be incorrectly configured, with potentially disastrous results. Because this MIB module reuses the IP tunnel MIB and nat MIB, the security considerations for these MIBs are also applicable to the DS-Lite MIB.

Unauthorized read access todsliteTunnelEndAddress, or any object in the dsliteBindRelationTable or dslitePortBindRelationTable would reveal information about the mapping information.

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.

12. References

12.1. Normative References

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

- [RFC2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2", RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", RFC 2580, April 1999.
- [RFC2863] McCloghrie, K. and F. Kastenholz. "The Interfaces Group MIB", RFC 2863, June 2000.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", RFC 3411, December 2002.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.
- [RFC4008] Rohit, R., Srisuresh, P., Raghunarayan, R., Pai, N., and Wang, C., "Definitions of Managed Objects for Network Address Translators (NAT)", RFC 4008, March 2005.
- [RFC4087] Thaler, D., "IP Tunnel MIB", RFC 4087, June 2005.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC6333, August 2011.

12.2. Informative References

- [I-D.ietf-softwire-gateway-init-ds-lite] Brockners, F., Gundavelli, S., Speicher, S., and D. Ward, "Gateway Initiated Dual-Stack Lite Deployment", draft-ietf-softwire-gateway-init-ds-lite-08 (work in progress), July 2011.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

13. Change Log [RFC Editor please remove]

draft-fu-softwire-dslite-mib-00, original version, 2011-05-04

draft-fu-softwire-dslite-mib-01, 01 version, 2011-07-11
draft-fu-softwire-dslite-mib-02, 02 version, 2011-08-27
draft-fu-softwire-dslite-mib-03, 03 version, 2012-02-22
draft-fu-softwire-dslite-mib-04, 04 version, 2012-04-24
draft-fu-softwire-dslite-mib-05, 05 version, 2012-05-28

Author's Addresses

Yu Fu
Huawei Technologies Co., Ltd
Huawei Building, 156 Beiqing Rd.,
Hai-Dian District, Beijing 100095
P.R. China
Email: eleven.fuyu@huawei.com

Sheng Jiang
Huawei Technologies Co., Ltd
Huawei Building, 156 Beiqing Rd.,
Hai-Dian District, Beijing 100095
P.R. China
Email: jiangsheng@huawei.com

Jiang Dong
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R. China
Email: dongjiang@csnet1.cs.tsinghua.edu.cn

Yuchi Chen
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R. China
Email: flashfoxmx@gmail.com

Softwire WG
Internet-Draft
Intended status: Standards Track
Expires: August 6, 2017

M. Boucadair
Orange
C. Qin
Cisco
C. Jacquenet
Orange
Y. Lee
Comcast
Q. Wang
China Telecom
February 2, 2017

Delivery of IPv4 Multicast Services to IPv4 Clients over an IPv6
Multicast Network
draft-ietf-softwire-dslite-multicast-18

Abstract

This document specifies a solution for the delivery of IPv4 multicast services to IPv4 clients over an IPv6 multicast network. The solution relies upon a stateless IPv4-in-IPv6 encapsulation scheme and uses an IPv6 multicast distribution tree to deliver IPv4 multicast traffic. The solution is particularly useful for the delivery of multicast service offerings to DS-Lite serviced customers.

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 August 6, 2017.

Copyright Notice

Copyright (c) 2017 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	3
1.1. Requirements Language	4
2. Terminology	4
3. Scope	5
4. Solution Overview	6
4.1. IPv4-Embedded IPv6 Prefixes	7
4.2. Multicast Distribution Tree Computation	7
4.3. Multicast Data Forwarding	8
5. IPv4/IPv6 Address Mapping	9
5.1. Prefix Assignment	9
5.2. Multicast Address Translation Algorithm	9
5.3. Textual Representation	10
5.4. Examples	10
6. Multicast B4 (mB4)	10
6.1. IGMP-MLD Interworking Function	10
6.2. Multicast Data Forwarding	11
6.3. Fragmentation	11
6.4. Host Built-in mB4 Function	12
6.5. Preserve the Scope	12
7. Multicast AFTR (mAFTR)	12
7.1. Routing Considerations	12
7.2. Processing PIM Messages	13
7.3. Switching from Shared Tree to Shortest Path Tree	14
7.4. Multicast Data Forwarding	14
7.5. Scope	14
8. Deployment Considerations	15
8.1. Other Operational Modes	15
8.1.1. The IPv6 DR is Co-Located with the mAFTR	15
8.1.2. The IPv4 DR is Co-Located with the mAFTR	15
8.2. Load Balancing	15
8.3. mAFTR Policy Configuration	15

8.4. Static vs. Dynamic PIM Triggering	16
9. Security Considerations	16
9.1. Firewall Configuration	16
10. Acknowledgments	16
11. IANA Considerations	17
12. References	17
12.1. Normative References	17
12.2. Informative References	18
Appendix A. Use Case: IPTV	19
Appendix B. Older Versions of Group Membership Management Protocols	19
Authors' Addresses	20

1. Introduction

DS-Lite [RFC6333] is an IPv4 address-sharing technique that enables operators to multiplex public IPv4 addresses while provisioning only IPv6 to users. A typical DS-Lite scenario is the delivery of an IPv4 service to an IPv4 user over an IPv6 network (denoted as a 4-6-4 scenario). [RFC6333] covers unicast services exclusively.

This document specifies a generic solution for the delivery of IPv4 multicast services to IPv4 clients over an IPv6 multicast network. The solution was developed with DS-Lite in mind (see more discussion below). The solution is however not limited to DS-Lite; it can be applied in other deployment contexts, such as [RFC7596][RFC7597].

If customers have to access IPv4 multicast-based services through a DS-Lite environment, Address Family Transition Router (AFTR) devices will have to process all the Internet Group Management Protocol (IGMP) Report messages [RFC2236] [RFC3376] that have been forwarded by the Customer Premises Equipment (CPE) into the IPv4-in-IPv6 tunnels. From that standpoint, AFTR devices are likely to behave as a replication point for downstream multicast traffic, and the multicast packets will be replicated for each tunnel endpoint that IPv4 receivers are connected to.

This kind of DS-Lite environment raises two major issues:

1. The IPv6 network loses the benefits of the multicast traffic forwarding efficiency because it is unable to deterministically replicate the data as close to the receivers as possible. As a consequence, the downstream bandwidth in the IPv6 network will be vastly consumed by sending multicast data over a unicast infrastructure.
2. The AFTR is responsible for replicating multicast traffic and forwarding it into each tunnel endpoint connecting IPv4 receivers

that have explicitly asked for the corresponding contents. This process may significantly consume the AFTR's resources and overload the AFTR.

This document specifies an extension to the DS-Lite model to deliver IPv4 multicast services to IPv4 clients over an IPv6 multicast-enabled network.

This document describes a stateless translation mechanism that supports either Source Specific Multicast (SSM) or Any Source Multicast (ASM) operation. The recommendation in Section 1 of [RFC4607] is that multicast services use SSM where possible; the operation of the translation mechanism is also simplified when SSM is used, e.g., considerations for placement of the IPv6 the Rendezvous Point (RP) are no longer relevant.

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

mPrefix64: a dedicated multicast IPv6 prefix for constructing IPv4-embedded IPv6 multicast addresses. mPrefix64 can be of two types: ASM_mPrefix64 used in Any Source Multicast (ASM) mode or SSM_mPrefix64 used in Source Specific Multicast (SSM) mode [RFC4607]. The size of this prefix is /96.

Note: "64" is used as an abbreviation for IPv6-IPv4 interconnection.

uPrefix64: a dedicated IPv6 unicast prefix for constructing IPv4-embedded IPv6 unicast addresses [RFC6052]. This prefix may be either the Well-Known Prefix (i.e., 64:ff9b::/96) or a Network-Specific Prefix (NSP).

Multicast AFTR (mAFTR): a functional entity which supports an IPv4-IPv6 multicast interworking function (refer to Figure 3). It receives and encapsulates the IPv4 multicast packets into IPv4-in-

IPv6 packets. Also, it behaves as the corresponding IPv6 multicast source for the encapsulated IPv4-in-IPv6 packets.

Multicast Basic Bridging BroadBand (mB4): a functional entity which supports an IGMP-MLD interworking function (refer to Section 6.1) that translates the IGMP messages into the corresponding Multicast Listener Discovery (MLD) messages, and sends the MLD messages to the IPv6 network. In addition, the mB4 decapsulates IPv4-in-IPv6 multicast packets.

PIMv4: refers to Protocol Independent Multicast (PIM) when deployed in an IPv4 infrastructure (i.e., IPv4 transport capabilities are used to exchange PIM messages).

PIMv6: refers to PIM when deployed in an IPv6 infrastructure (i.e., IPv6 transport capabilities are used to exchange PIM messages).

Host portion of the MLD protocol: refers to the part of MLD that applies to all multicast address listeners (Section 6 of [RFC3810]). As a reminder, MLD specifies separate behaviors for multicast address listeners (i.e., hosts or routers that listen to multicast packets) and multicast routers.

Router portion of the IGMP protocol: refers to the part of IGMP that is performed by multicast routers (Section 6 of [RFC3376]).

DR: refers to the Designated Router as defined in [RFC7761].

3. Scope

This document focuses only on the subscription to IPv4 multicast groups and the delivery of IPv4-formatted content to IPv4 receivers over an IPv6-only network. In particular, only the following case is covered:

IPv4 receivers access IPv4 multicast contents over IPv6-only multicast-enabled networks.

This document does not cover the source/receiver heuristics, where IPv4 receivers can also behave as IPv4 multicast sources. This document assumes that hosts behind the mB4 are IPv4 multicast receivers only. Also, the document covers host built-in mB4 function.

4. Solution Overview

In the DS-Lite specification [RFC6333], an IPv4-in-IPv6 tunnel is used to carry bidirectional IPv4 unicast traffic between a B4 and an AFTR. The solution specified in this document provides an IPv4-in-IPv6 encapsulation scheme to deliver unidirectional IPv4 multicast traffic from an mAFTR to an mB4.

An overview of the solution is provided in this section which is intended as an introduction to how it works, but is not normative. For the normative specifications of the two new functional elements: mB4 and mAFTR (Figure 1), refer to Sections 6 and 7.

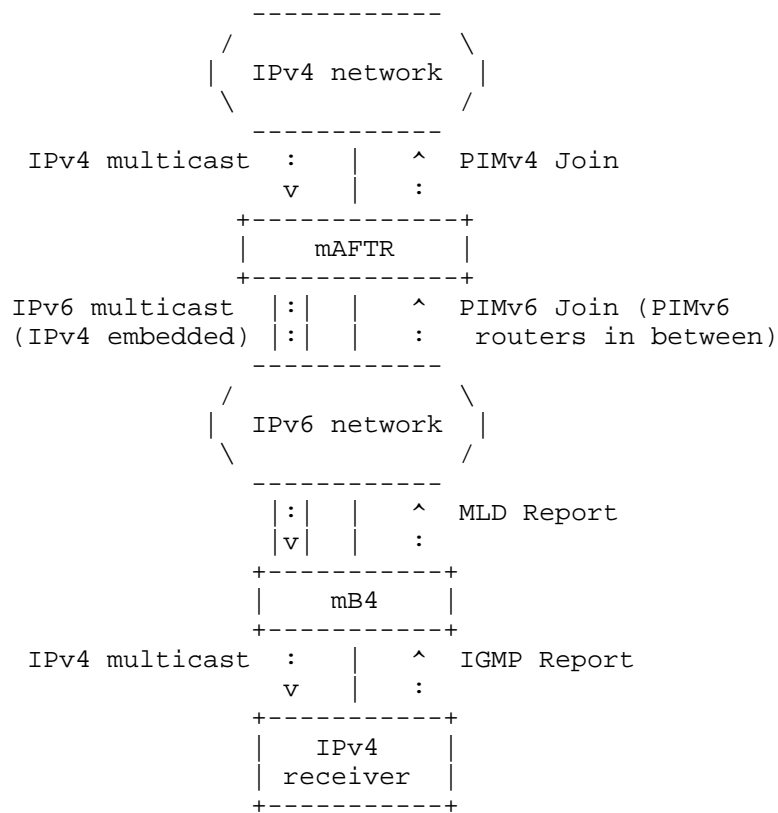


Figure 1: Functional Architecture

4.1. IPv4-Embedded IPv6 Prefixes

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 the mAFTR and the mB4 elements, both of which contribute to the computation and the maintenance of the IPv6 multicast distribution tree that extends the IPv4 multicast distribution tree into the IPv6 multicast network. The IPv4/IPv6 address mapping is stateless.

The mAFTR and the mB4 use mPrefix64 to convert an IPv4 multicast address (G4) into an IPv4-embedded IPv6 multicast address (G6). The mAFTR and the mB4 use uPrefix64 to convert an IPv4 source address (S4) into an IPv4-embedded IPv6 address (S6). The mAFTR and the mB4 must use the same mPrefix64 and uPrefix64, and also run the same algorithm for building IPv4-embedded IPv6 addresses. Refer to Section 5 for more details about the address mapping.

4.2. Multicast Distribution Tree Computation

When an IPv4 receiver connected to the device that embeds the mB4 capability wants to subscribe to an IPv4 multicast group, it sends an IGMP Report message towards the mB4. The mB4 creates the IPv6 multicast group (G6) address using mPrefix64 and the original IPv4 multicast group address. If the receiver sends a source-specific IGMPv3 Report message, the mB4 will create the IPv6 source address (S6) using uPrefix64 and the original IPv4 source address.

The mB4 uses the G6 (and both S6 and G6 in SSM) to create the corresponding MLD Report message. The mB4 sends the Report message towards the IPv6 network. The PIMv6 Designated Router receives the MLD Report message and sends the PIMv6 Join message to join the IPv6 multicast distribution tree. It can send either PIMv6 Join (*,G6) in ASM or PIMv6 Join (S6,G6) in SSM to the mAFTR.

The mAFTR acts as the IPv6 DR to which the uPrefix64-derived S6 is connected. The mAFTR will receive the source-specific PIMv6 Join message (S6,G6) from the IPv6 multicast network. If the mAFTR is the Rendezvous Point (RP) of G6, it will receive the any-source PIMv6 Join message (*,G6) from the IPv6 multicast network. If the mAFTR is not the RP of G6, it will send the PIM Register message to the RP of G6 located in the IPv6 multicast network. For the sake of simplicity, it is recommended to configure the mAFTR as the RP for the IPv4-embedded IPv6 multicast groups it manages; no registration procedure is required under this configuration.

When the mAFTR receives the PIMv6 Join message (*,G6), it will extract the IPv4 multicast group address (G4). If the mAFTR is the

RP of G4 in the IPv4 multicast network, it will create a (*,G4) entry (if such entry does not already exist) in its own IPv4 multicast routing table. If the mAFTR is not the RP of G4, it will send the corresponding PIMv4 Join message (*,G4) towards the RP of G4 in the IPv4 multicast network.

When the mAFTR receives the PIMv6 Join message (S6,G6), it will extract the IPv4 multicast group address (G4) and IPv4 source address (S4) and send the corresponding (S4,G4) PIMv4 Join message directly to the IPv4 source.

A branch of the multicast distribution tree is thus constructed, comprising both an IPv4 part (from the mAFTR upstream) and an IPv6 part (from mAFTR downstream towards the mB4).

The mAFTR advertises the route of uPrefix64 with an IPv6 Interior Gateway Protocol (IGP), so as to represent the IPv4-embedded IPv6 source in the IPv6 multicast network, and to allow IPv6 routers to run the Reverse Path Forwarding (RPF) check procedure on incoming multicast traffic. Injecting internal /96 routes is not problematic given the recommendation in [RFC7608] that requires that forwarding processes must be designed to process prefixes of any length up to /128.

4.3. Multicast Data Forwarding

When the mAFTR receives an IPv4 multicast packet, it will encapsulate the packet into an IPv6 multicast packet using the IPv4-embedded IPv6 multicast address as the destination address and an IPv4-embedded IPv6 unicast address as the source address. The encapsulated IPv6 multicast packet will be forwarded down the IPv6 multicast distribution tree and the mB4 will eventually receive the packet.

The IPv6 multicast network treats the IPv4-in-IPv6 encapsulated multicast packets as native IPv6 multicast packets. The IPv6 multicast routers use the outer IPv6 header to make their forwarding decisions.

When the mB4 receives the IPv6 multicast packet (to G6) derived by mPrefix64, it decapsulates it and forwards the original IPv4 multicast packet towards the receivers subscribing to G4.

Note: At this point, only IPv4-in-IPv6 encapsulation is defined; however, other types of encapsulation could be defined in the future.

5. IPv4/IPv6 Address Mapping

5.1. Prefix Assignment

A dedicated IPv6 multicast prefix (mPrefix64) is provisioned to the mAFTR and the mB4. The mAFTR and the mB4 use the mPrefix64 to form an IPv6 multicast group address from an IPv4 multicast group address. The mPrefix64 can be of two types: ASM_mPrefix64 (a mPrefix64 used in ASM mode) or SSM_mPrefix64 (a mPrefix64 used in SSM mode). The mPrefix64 MUST be derived from the corresponding IPv6 multicast address space (e.g., the SSM_mPrefix64 must be in the range of multicast address space specified in [RFC4607]).

The IPv6 part of the multicast distribution tree can be seen as an extension of the IPv4 part of the multicast distribution tree. The IPv4 source address MUST be mapped to an IPv6 source address. An IPv6 unicast prefix (uPrefix64) is provisioned to the mAFTR and the mB4. The mAFTR and the mB4 use the uPrefix64 to form an IPv6 source address from an IPv4 source address as specified in [RFC6052]. The uPrefix-formed IPv6 source address will represent the original IPv4 source in the IPv6 multicast network. The uPrefix64 MUST be derived from the IPv6 unicast address space.

The multicast address translation MUST follow the algorithm defined in Section 5.2.

The mPrefix64 and uPrefix64 can be configured in the mB4 using a variety of methods, including an out-of-band mechanism, manual configuration, or a dedicated provisioning protocol (e.g., using DHCPv6 [I-D.ietf-software-multicast-prefix-option]).

The stateless translation mechanism described in Section 5 does not preclude use of Embedded-RP [RFC3956][RFC7371].

5.2. Multicast Address Translation Algorithm

IPv4-embedded IPv6 multicast addresses are composed according to the following algorithm:

- o Concatenate the mPrefix64 96 bits and the 32 bits of the IPv4 address to obtain a 128-bit address.

The IPv4 multicast addresses are extracted from the IPv4-embedded IPv6 multicast addresses according to the following algorithm:

- o If the multicast address has a pre-configured mPrefix64, extract the last 32 bits of the IPv6 multicast address.

An IPv4 source is represented in the IPv6 realm with its IPv4-converted IPv6 address [RFC6052].

5.3. Textual Representation

The embedded IPv4 address in an IPv6 multicast address is included in the last 32 bits; therefore, dotted decimal notation can be used.

5.4. Examples

Group address mapping example:

mPrefix64	IPv4 address	IPv4-Embedded IPv6 address
ff0x::db8:0:0/96	233.252.0.1	ff0x::db8:233.252.0.1

Source address mapping example when a /96 is used:

uPrefix64	IPv4 address	IPv4-Embedded IPv6 address
2001:db8::/96	192.0.2.33	2001:db8::192.0.2.33

IPv4 and IPv6 addresses used in this example are derived from the IPv4 and IPv6 blocks reserved for documentation, as per [RFC6676]. The unicast IPv4 address of the above example is derived from the documentation address block defined in [RFC6890].

6. Multicast B4 (mB4)

6.1. IGMP-MLD Interworking Function

The IGMP-MLD Interworking Function combines the IGMP/MLD Proxying function and the address synthesizing operations. The IGMP/MLD Proxying function is specified in [RFC4605]. The address translation is stateless and MUST follow the address mapping specified in Section 5.

The mB4 performs the host portion of the MLD protocol on the upstream interface. The composition of IPv6 membership in this context is constructed through address synthesizing operations and MUST synchronize with the membership database maintained in the IGMP domain. MLD messages are sent natively to the directly connected IPv6 multicast routers (it will be processed by the PIM DR). The mB4

also performs the router portion of the IGMP protocol on the downstream interface(s). Refer to [RFC4605] for more details.

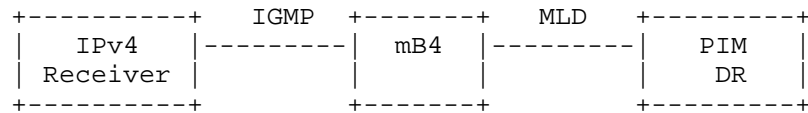


Figure 2: IGMP-MLD Interworking

If SSM is deployed, the mB4 MUST construct the IPv6 source address (or retrieve the IPv4 source address) using the uPrefix64. The mB4 MAY create a membership database which associates the IPv4-IPv6 multicast groups with the interfaces (e.g., WLAN and Wired Ethernet) facing IPv4 multicast receivers.

6.2. Multicast Data Forwarding

When the mB4 receives an IPv6 multicast packet, it MUST check the group address and the source address. If the IPv6 multicast group prefix is mPrefix64 and the IPv6 source prefix is uPrefix64, the mB4 MUST decapsulate the IPv6 header [RFC2473]; the decapsulated IPv4 multicast packet will be forwarded through each relevant interface following standard IPv4 multicast forwarding procedure. Otherwise, the mB4 MUST silently drop the packet.

As an illustration, if a packet is received from source 2001:db8::192.0.2.33 and needs to be forwarded to group ff3x:20:2001:db8::233.252.0.1, the mB4 decapsulates it into an IPv4 multicast packet using 192.0.2.33 as the IPv4 source address and using 233.252.0.1 as the IPv4 destination multicast group. This example assumes that the mB4 is provisioned with uPrefix64 (2001:db8::/96) and mPrefix64 (ff3x:20:2001:db8::/96).

6.3. Fragmentation

Encapsulating IPv4 multicast packets into IPv6 multicast packets that will be forwarded by the mAFTR towards the mB4 along the IPv6 multicast distribution tree reduces the effective MTU size by the size of an IPv6 header. In this specification, the data flow is unidirectional from the mAFTR to the mB4. The mAFTR MUST fragment the oversized IPv6 packet after the encapsulation into two IPv6 packets. The mB4 MUST reassemble the IPv6 packets, decapsulate the IPv6 header, and forward the IPv4 packet to the hosts that have subscribed to the corresponding multicast group. Further considerations about fragmentation issues are documented in Sections 5.3 and 6.3 of [RFC6333].

6.4. Host Built-in mB4 Function

If the mB4 function is implemented in the host which is directly connected to an IPv6-only network, the host MUST implement the behaviors specified in Sections 6.1, 6.2, and 6.3. The host MAY optimize the implementation to provide an Application Programming Interface (API) or kernel module to skip the IGMP-MLD Interworking Function. Optimization considerations are out of scope of this specification.

6.5. Preserve the Scope

When several mPrefix64s are available, if each enclosed IPv4-embedded IPv6 multicast prefix has a distinct scope, the mB4 MUST select the appropriate IPv4-embedded IPv6 multicast prefix whose scope matches the IPv4 multicast address used to synthesize an IPv4-embedded IPv6 multicast address (specific mappings are listed in Section 8 of [RFC2365]). Mapping is achieved such that the scope of the selected IPv6 multicast prefix does not exceed the original IPv4 multicast scope. If the mB4 is instructed to preserve the scope but no IPv6 multicast prefix that matches the IPv4 multicast scope is found, IPv6 multicast address mapping SHOULD fail.

The mB4 MAY be configured to not preserve the scope when enforcing the address translation algorithm.

Consider that an mB4 is configured with two mPrefix64s ff0e::db8:0:0/96 (Global scope) and ff08::db8:0:0/96 (Organization scope). If the mB4 receives an IGMP report from an IPv4 receiver to subscribe to 233.252.0.1, it checks which mPrefix64 to use in order to preserve the scope of the requested IPv4 multicast group. In this example, given that 233.252.0.1 is intended for global use, the mB4 creates the IPv6 multicast group (G6) address using ff0e::db8:0:0/96 and the original IPv4 multicast group address (233.252.0.1): ff0e::db8:233.252.0.1.

7. Multicast AFTR (mAFTR)

7.1. Routing Considerations

The mAFTR is responsible for interconnecting the IPv4 multicast distribution tree with the corresponding IPv6 multicast distribution tree. The mAFTR MUST use the uPrefix64 to build the IPv6 source addresses of the multicast group address derived from mPrefix64. In other words, the mAFTR MUST be the multicast source whose address is derived from uPrefix64.

The mAFTR MUST advertise the route towards uPrefix64 with the IPv6 IGP. This is needed by the IPv6 multicast routers so that they acquire the routing information to discover the source.

7.2. Processing PIM Messages

The mAFTR MUST interwork PIM Join/Prune messages for (*,G6) and (S6,G6) on their corresponding (*,G4) and (S4,G4). The following text specifies the expected behavior of the mAFTR for PIM Join messages.

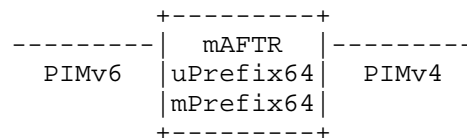


Figure 3: PIMv6-PIMv4 Interworking Function

The mAFTR contains two separate Tree Information Bases (TIBs): the IPv4 Tree Information Base (TIB4) and the IPv6 Tree Information Base (TIB6), which are bridged by one IPv4-in-IPv6 virtual interface. It should be noted that TIB implementations may vary (e.g., some may rely upon a single integrated TIB without any virtual interface), but they should follow this specification for the sake of global and functional consistency.

When an mAFTR receives a PIMv6 Join message (*,G6) with an IPv6 multicast group address (G6) that is derived from the mPrefix64, it MUST check its IPv6 Tree Information Base (TIB6). If there is an entry for this G6 address, it MUST check whether the interface through which the PIMv6 Join message has been received is in the outgoing interface (oif) list. If not, the mAFTR MUST add the interface to the oif list. If there is no entry in the TIB6, the mAFTR MUST create a new entry (*,G6) for the multicast group. Whether or not the IPv4-in-IPv6 virtual interface is set as the incoming interface of the newly created entry is up to the implementation but it should comply with the mAFTR's multicast data forwarding behavior, see Section 7.4.

The mAFTR MUST extract the IPv4 multicast group address (G4) from the IPv4-embedded IPv6 multicast address (G6) contained in the PIMv6 Join message. The mAFTR MUST check its IPv4 Tree Information Base (TIB4). If there is an entry for G4, it MUST check whether the IPv4-in-IPv6 virtual interface is in the outgoing interface list. If not, the mAFTR MUST add the interface to the oif list. If there is no entry for G4, the mAFTR MUST create a new (*,G4) entry in its TIB4 and

initiate the procedure for building the shared tree in the IPv4 multicast network without any additional requirement.

If the mAFTR receives a source-specific Join message, the (S6,G6) is processed rather than (*,G6). The procedures of processing (S6,G6) and (*,G6) are almost the same. Differences have been detailed in [RFC7761].

7.3. Switching from Shared Tree to Shortest Path Tree

When the mAFTR receives the first IPv4 multicast packet, it may extract the source address (S4) from the packet and send an Explicit PIMv4 (S4,G4) Join message directly to S4. The mAFTR switches from the shared Rendezvous Point Tree (RPT) to the Shortest Path Tree (SPT) for G4.

For IPv6 multicast routers to switch to the SPT, there is no new requirement. IPv6 multicast routers may send an Explicit PIMv6 Join to the mAFTR once the first (S6,G6) multicast packet arrives from upstream multicast routers.

7.4. Multicast Data Forwarding

When the mAFTR receives an IPv4 multicast packet, it checks its TIB4 to find a matching entry and then forwards the packet to the interface(s) listed in the outgoing interface list. If the IPv4-in-IPv6 virtual interface also belongs to this list, the packet is encapsulated with the mPrefix64-derived and uPrefix64-derived IPv4-embedded IPv6 addresses to form an IPv6 multicast packet [RFC2473]. Then another lookup is made by the mAFTR to find a matching entry in the TIB6. Whether the RPF check for the second lookup is performed or not is up to the implementation and is out of the scope of this document. The IPv6 multicast packet is then forwarded along the IPv6 multicast distribution tree, based upon the outgoing interface list of the matching entry in the TIB6.

As an illustration, if a packet is received from source 192.0.2.33 and needs to be forwarded to group 233.252.0.1, the mAFTR encapsulates it into an IPv6 multicast packet using ff3x:20:2001:db8::233.252.0.1 as the IPv6 destination multicast group and using 2001:db8::192.0.2.33 as the IPv6 source address.

7.5. Scope

The Scope field of IPv4-in-IPv6 multicast addresses should be valued accordingly (e.g, to "E" for Global scope) in the deployment environment. This specification does not discuss the scope value that should be used.

The considerations in Section 6.5 are to be followed by the mAFTR.

8. Deployment Considerations

8.1. Other Operational Modes

8.1.1. The IPv6 DR is Co-Located with the mAFTR

The mAFTR can embed the MLD Querier function (as well as the PIMv6 DR) for optimization purposes. When the mB4 sends a MLD Report message to this mAFTR, the mAFTR should process the MLD Report message that contains the IPv4-embedded IPv6 multicast group address and then send the corresponding PIMv4 Join message (Figure 4).

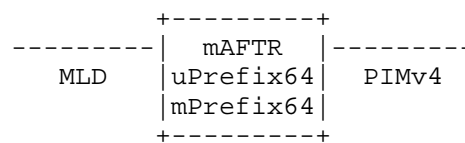


Figure 4: MLD-PIMv4 Interworking Function

Discussions about the location of the mAFTR capability and related ASM or SSM multicast design considerations are out of the scope of this document.

8.1.2. The IPv4 DR is Co-Located with the mAFTR

If the mAFTR is co-located with the IPv4 DR connected to the original IPv4 source, it may simply use the uPrefix64 and mPrefix64 prefixes to build the IPv4-embedded IPv6 multicast packets, and the sending of PIMv4 Join messages becomes unnecessary.

8.2. Load Balancing

For robustness 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.

8.3. mAFTR Policy Configuration

The mAFTR may be configured with a list of IPv4 multicast groups and sources. Only multicast flows bound to the configured addresses should be handled by the mAFTR. Otherwise, packets are silently dropped.

8.4. Static vs. Dynamic PIM Triggering

To optimize the usage of network resources in current deployments, all multicast streams are conveyed in the core network while only the most popular ones are forwarded in the aggregation/access networks (static mode). Less popular streams are forwarded in the access network upon request (dynamic mode). Depending on the location of the mAFTR in the network, two modes can be envisaged: static and dynamic.

Static Mode: the mAFTR is configured to instantiate permanent (S6,G6) and (*,G6) entries in its TIB6 using a pre-configured (S4,G4) list.

Dynamic Mode: the instantiation or withdrawal of (S6,G6) or (*,G6) entries is triggered by the receipt of PIMv6 messages.

9. Security Considerations

Besides multicast scoping considerations (see Section 6.5 and Section 7.5), 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 [RFC7761].

Unlike solutions that map IPv4 multicast flows to IPv6 unicast flows, this document does not exacerbate Denial-of-Service (DoS) attacks.

An mB4 SHOULD be provided with appropriate configuration information to preserve the scope of a multicast message when mapping an IPv4 multicast address into an IPv4-embedded IPv6 multicast address and vice versa.

9.1. Firewall Configuration

The CPE that embeds the mB4 function SHOULD be configured to accept incoming MLD messages and traffic forwarded to multicast groups subscribed by receivers located in the customer premises.

10. Acknowledgments

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

Many thanks to Ian Farrer for the review.

Thanks to Zhen Cao, Tim Chown, Francis Dupont, Jouni Korhonen, and Stig Venaas for the directorates review.

11. IANA Considerations

This document includes no request to IANA.

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2365] Meyer, D., "Administratively Scoped IP Multicast", BCP 23, RFC 2365, DOI 10.17487/RFC2365, July 1998, <<http://www.rfc-editor.org/info/rfc2365>>.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <<http://www.rfc-editor.org/info/rfc2473>>.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <<http://www.rfc-editor.org/info/rfc3376>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<http://www.rfc-editor.org/info/rfc3810>>.
- [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, DOI 10.17487/RFC4605, August 2006, <<http://www.rfc-editor.org/info/rfc4605>>.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <<http://www.rfc-editor.org/info/rfc4607>>.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, DOI 10.17487/RFC6052, October 2010, <<http://www.rfc-editor.org/info/rfc6052>>.

- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <<http://www.rfc-editor.org/info/rfc6333>>.
- [RFC7608] Boucadair, M., Petrescu, A., and F. Baker, "IPv6 Prefix Length Recommendation for Forwarding", BCP 198, RFC 7608, DOI 10.17487/RFC7608, July 2015, <<http://www.rfc-editor.org/info/rfc7608>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<http://www.rfc-editor.org/info/rfc7761>>.

12.2. Informative References

- [I-D.ietf-softwire-multicast-prefix-option] Boucadair, M., Qin, J., Tsou, T., and X. Deng, "DHCPv6 Option for IPv4-Embedded Multicast and Unicast IPv6 Prefixes", draft-ietf-softwire-multicast-prefix-option-13 (work in progress), February 2017.
- [RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, DOI 10.17487/RFC2236, November 1997, <<http://www.rfc-editor.org/info/rfc2236>>.
- [RFC3956] Savola, P. and B. Haberman, "Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address", RFC 3956, DOI 10.17487/RFC3956, November 2004, <<http://www.rfc-editor.org/info/rfc3956>>.
- [RFC6676] Venaas, S., Parekh, R., Van de Velde, G., Chown, T., and M. Eubanks, "Multicast Addresses for Documentation", RFC 6676, DOI 10.17487/RFC6676, August 2012, <<http://www.rfc-editor.org/info/rfc6676>>.
- [RFC6890] Cotton, M., Vegoda, L., Bonica, R., Ed., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, DOI 10.17487/RFC6890, April 2013, <<http://www.rfc-editor.org/info/rfc6890>>.
- [RFC7371] Boucadair, M. and S. Venaas, "Updates to the IPv6 Multicast Addressing Architecture", RFC 7371, DOI 10.17487/RFC7371, September 2014, <<http://www.rfc-editor.org/info/rfc7371>>.

- [RFC7596] Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture", RFC 7596, DOI 10.17487/RFC7596, July 2015, <<http://www.rfc-editor.org/info/rfc7596>>.
- [RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <<http://www.rfc-editor.org/info/rfc7597>>.

Appendix A. Use Case: IPTV

IPTV generally includes two categories of service offerings:

- o Video on Demand (VoD) that unicast video content to receivers.
- o Multicast live TV broadcast services.

Two types of provider are involved in the delivery of this service:

- o Content Providers, who usually own the contents that is multicast to receivers. Content providers may contractually define an agreement with network providers to deliver contents to receivers.
- o Network Providers, who provide network connectivity services (e.g., network providers are responsible for carrying multicast flows from head-ends to receivers).

Note that some contract agreements prevent a network provider from altering the content as sent by the content provider for various reasons. Depending on these contract agreements, multicast streams should be delivered unaltered to the requesting users.

Many 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 to support IPv6. As a consequence, IPv4 service continuity must be guaranteed during the transition period, including the delivery of multicast services such as Live TV Broadcasting to users.

Appendix B. Older Versions of Group Membership Management Protocols

Given the multiple versions of group membership management protocols, mismatch issues may arise at the mB4 (refer to Section 6.1).

If IGMPv2 operates on the IPv4 receivers while MLDv2 operates on the MLD Querier, or if IGMPv3 operates on the IPv4 receivers while MLDv1 operates on the MLD Querier, the version mismatch issue will be encountered. To solve this problem, the mB4 should perform the router portion of IGMP which is similar to the corresponding MLD version (IGMPv2 as of MLDv1, or IGMPv3 as of MLDv2) operating in the IPv6 domain. Then, the protocol interaction approach specified in Section 7 of [RFC3376] can be applied to exchange signaling messages with the IPv4 receivers on which the different version of IGMP is operating.

Note that the support of IPv4 SSM requires MLDv2 to be enabled in the IPv6 network.

Authors' Addresses

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Chao Qin
Cisco
Shanghai
P.R. China

Email: jacni@jacni.com

Christian Jacquenet
Orange
Rennes 35000
France

Email: christian.jacquenet@orange.com

Yiu L. Lee
Comcast
United States of America

Email: yiulee@cable.comcast.com
URI: <http://www.comcast.com>

Qian Wang
China Telecom
P.R. China

Phone: +86 10 58502462
Email: 13301168516@189.cn

Softwire WG
Internet-Draft
Intended status: Standards Track
Expires: December 10, 2019

M. Xu
Y. Cui
J. Wu
Tsinghua University
S. Yang
Shenzhen University
C. Metz
Cisco Systems
June 8, 2019

IPv4 Multicast over an IPv6 Multicast in Softwire Mesh Network
draft-ietf-softwire-mesh-multicast-25

Abstract

During the transition to IPv6, there will be scenarios where a backbone network internally running one IP address family (referred to as the internal IP or I-IP family), connects client networks running another IP address family (referred to as the external IP or E-IP family). In such cases, the I-IP backbone needs to offer both unicast and multicast transit services to the client E-IP networks.

This document describes a mechanism for supporting multicast across backbone networks where the I-IP and E-IP protocol families differ. The document focuses on IPv4-over-IPv6 scenario, due to lack of real-world use cases for IPv6-over-IPv4 scenario.

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 <https://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, 2019.

Copyright Notice

Copyright (c) 2019 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 (<https://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	3
2. Requirements Language	5
3. Terminology	5
4. Scope	6
5. Mesh Multicast Mechanism	7
5.1. Mechanism Overview	8
5.2. Group Address Mapping	8
5.3. Source Address Mapping	9
5.4. Routing Mechanism	9
6. Control Plane Functions of AFBR	10
6.1. E-IP (*,G) and (S,G) State Maintenance	10
6.2. I-IP (S',G') State Maintenance	10
6.3. E-IP (S,G,rpt) State Maintenance	11
6.4. Inter-AFBR Signaling	11
6.5. SPT Switchover	13
6.6. Other PIM Message Types	13
6.7. Other PIM States Maintenance	13
7. Data Plane Functions of the AFBR	14
7.1. Process and Forward Multicast Data	14
7.2. TTL or Hop Count	14
7.3. Fragmentation	14
8. Packet Format and Translation	14
9. Softwire Mesh Multicast Encapsulation	15
10. Security Considerations	16
11. IANA Considerations	16
12. Normative References	16
Appendix A. Acknowledgements	18
Authors' Addresses	18

1. Introduction

During the transition to IPv6, there will be scenarios where a backbone network internally running one IP address family (referred to as the internal IP or I-IP family), connects client networks running another IP address family (referred to as the external IP or E-IP family).

One solution is to leverage the multicast functions inherent in the I-IP backbone to efficiently forward client E-IP multicast packets inside an I-IP core tree. The I-IP tree is rooted at one or more ingress Address Family Border Routers (AFBRs) [RFC5565] and branches out to one or more egress AFBRs.

[RFC4925] outlines the requirements for the softwire mesh scenario and includes support for multicast traffic. It is likely 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 the client E-IP source-rooted or shared tree to traverse the I-IP backbone network.

This could be accomplished by re-using the multicast VPN approach outlined in [RFC6513]. MVPN-like schemes can support the softwire mesh scenario and achieve a "many-to-one" mapping between the E-IP client multicast trees and the 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 too many (S,G) states, which is source specific states related with a specified multicast group [RFC7761][RFC7899]. Aggregation at the edge contains the (S,G) states for customer's VPNs and these need to be maintained by the network operator. The disadvantage of this approach is the possibility of inefficient bandwidth and resource utilization when multicast packets are delivered to a receiving AFBR with no attached E-IP receivers.

[RFC8114] provides a solution for delivering IPv4 multicast services over an IPv6 network. But it mainly focuses on the DS-lite [RFC6333] scenario, where IPv4 addresses assigned by a broadband service provider are shared among customers. This document describes a detailed solution for the IPv4-over-IPv6 softwire mesh scenario, where client networks run IPv4 and the backbone network runs IPv6.

Internet-style multicast is somewhat different to the [RFC8114] scenario in that the trees are source-rooted and relatively sparse. The need for multicast aggregation at the edge (where many customer multicast trees are mapped into one or more backbone multicast trees)

does not exist and to date has not been identified. Thus the need for alignment between the E-IP and I-IP multicast mechanisms emerges.

[RFC5565] describes the "Softwire Mesh Framework". This document provides a more detailed description of how one-to-one mapping schemes ([RFC5565], Section 11.1) for IPv4-over-IPv6 multicast can be achieved.

Figure 1 shows an example of how a softwire mesh network can support multicast traffic. 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 a single multicast group residing in different client E-IP networks. In the case of shared trees, the E-IP sources, receivers and rendezvous points (RPs) might be located in different client E-IP networks. In the simplest case, a single operator manages the resources of the I-IP core, although the inter-operator case is also possible and so not precluded.

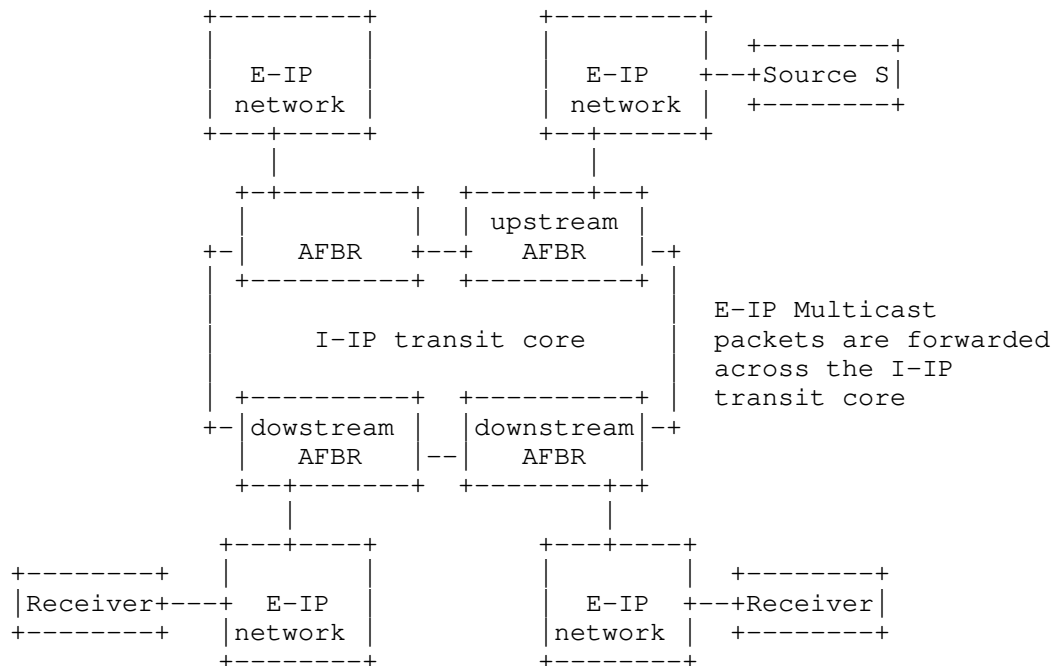


Figure 1: Software Mesh Multicast Framework

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology

Terminology used in this document:

- o Address Family Border Router (AFBR) - A router interconnecting two or more networks using different IP address families. Additionally, in the context of software 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: An AFBR that is closer to the source of a multicast data flow.
- o Downstream AFBR: An AFBR that is closer to a receiver of a multicast data flow.
- o I-IP (Internal IP): This refers to IP address family that is supported by the core network. In this document, the I-IP is IPv6.
- o E-IP (External IP): This refers to the IP address family that is supported by the client network(s) attached to the I-IP transit core. In this document, the E-IP is IPv4.
- o I-IP core tree: A 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 (in this document, we focus on IP multicast) operating exclusively inside the I-IP core network. An I-IP core tree is used to forward 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 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.
- o uPrefix64: The /96 unicast IPv6 prefix for constructing an IPv4-embedded IPv6 unicast address [RFC8114].
- o mPrefix64: The /96 multicast IPv6 prefix for constructing an IPv4-embedded IPv6 multicast address [RFC8114].
- o PIMv4, PIMv6: refer to [RFC8114].
- o Inter-AFBR signaling: A mechanism used by downstream AFBRs to send PIMv6 messages to the upstream AFBR.

4. Scope

This document focuses on the IPv4-over-IPv6 scenario, as shown in the following diagram:

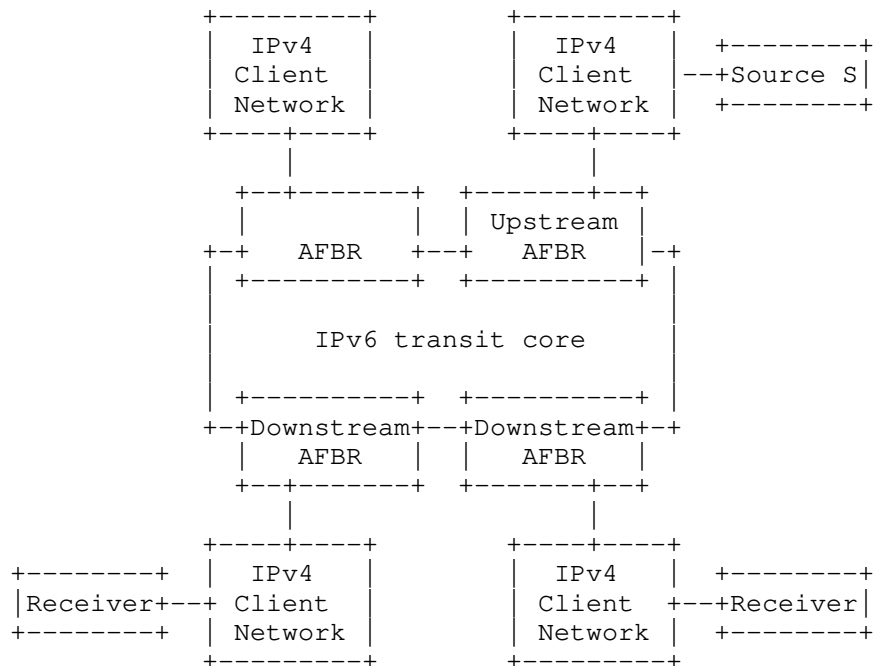


Figure 2: IPv4-over-IPv6 Scenario

In Figure 2, the E-IP client networks run IPv4 and the I-IP core runs IPv6.

Because of the much larger IPv6 group address space, the client E-IP tree can be mapped to a specific I-IP 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 support native IPv6 services and applications but in many cases, support for legacy IPv4 unicast and multicast services will also need to be accommodated.

5. Mesh Multicast Mechanism

5.1. Mechanism Overview

Routers in the client E-IP networks have routes to all other client E-IP networks. Through PIMv4 messages, E-IP hosts and routers have discovered or learnt of (S,G) or (*,G) [RFC7761] IPv4 addresses. Any I-IP multicast state instantiated in the core is referred to as (S',G') or (*,G') and is separated from E-IP multicast state.

Suppose a downstream AFBR receives an E-IP PIM Join/Prune message from the E-IP network for either an (S,G) tree or a (*,G) tree. The AFBR translates the PIMv4 message into an PIMv6 message with the latter being directed towards the I-IP IPv6 address of the upstream AFBR. When the PIMv6 message arrives at the upstream AFBR, it is translated back into an PIMv4 message. The result of these actions is the construction of E-IP trees and a corresponding I-IP tree in the I-IP network. An example of the packet format and translation is provided in Section 8.

In this case, it is incumbent upon the AFBRs to perform PIM message conversions in the control plane and IP group address conversions or mappings in the data plane. The AFBRs perform an algorithmic, one-to-one mapping of IPv4-to-IPv6.

5.2. Group Address Mapping

A simple algorithmic mapping between IPv4 multicast group addresses and IPv6 group addresses is performed. Figure 3 is provided as a reminder of the format:

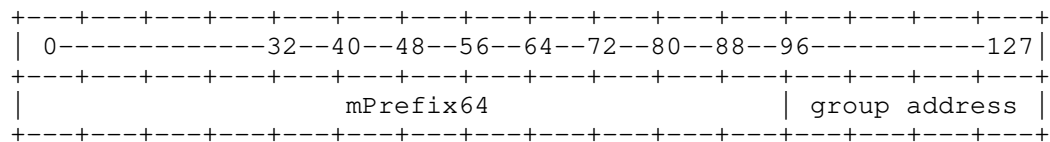


Figure 3: IPv4-Embedded IPv6 Multicast Address Format

An IPv6 multicast prefix (mPrefix64) is provisioned on each AFBR. AFBRs will prepend the prefix to an IPv4 multicast group address when translating it to an IPv6 multicast group address.

The construction of the mPrefix64 for Source-Specific Multicast (SSM) is the same as the construction of the mPrefix64 described in Section 5 of [RFC8114].

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 an IPv4 multicast address. The group address translation algorithm can be referred in Section 5.2 of [RFC8114].

5.3. Source Address Mapping

There are two kinds of multicast: Any-Source Multicast (ASM) and SSM. Considering that the I-IP network and E-IP network may support different kinds of multicast, the source address translation rules needed to support all possible scenarios may become very complex. But since SSM can be implemented with a strict subset of the PIM-SM protocol mechanisms [RFC7761], we can treat the I-IP core as SSM-only to make it as simple as possible. There then remain only two scenarios to be discussed in detail:

- o E-IP network supports SSM

One possible way to make sure that the translated PIMv6 message reaches upstream AFBR is to set S' to a virtual IPv6 address that leads to the upstream AFBR. The unicast address translation should be achieved according to [RFC6052]

- o E-IP network supports ASM

The (S,G) source list entry and the (*,G) source list entry differ only in that the latter has both the WildCard (WC) and RPT bits of the Encoded-Source-Address set, while with the former, the bits are cleared (See Section 4.9.5.1 of [RFC7761]). As a result, the source list entries in (*,G) messages can be translated into source list entries in (S',G') messages by clearing both the WC and RPT bits at downstream AFBRs, and vice-versa for the reverse translation at upstream AFBRs.

5.4. Routing Mechanism

With mesh multicast, PIMv6 messages originating from a downstream AFBR need to be propagated to the correct upstream AFBR, and every AFBR needs the /96 prefix in "IPv4-Embedded IPv6 Source Address Format" [RFC6052].

To achieve this, every AFBR MUST announce the address of one of its E-IPv4 interfaces in the "v4" field [RFC6052] alongside the corresponding uPrefix46. The announcement MUST be sent to the other AFBRs through MBGP [RFC4760]. Every uPrefix64 that an AFBR announces

MUST be unique. "uPrefix64" is an IPv6 prefix, and the distribution mechanism is the same as the traditional mesh unicast scenario.

As the "v4" field is an E-IP address, and BGP messages are not tunneled through softwires or any other mechanism specified in [RFC5565], AFBRs MUST be able to transport and encode/decode BGP messages that are carried over the I-IP, and whose NLRI and NH are of the E-IP address family.

In this way, when a downstream AFBR receives an E-IP PIM (S,G) message, it can translate this message into (S',G') by looking up the IP address of the corresponding AFBR's E-IP interface. Since the uPrefix64 of S' is unique, and is known to every router in the I-IP network, the translated message will be forwarded to the corresponding upstream AFBR, and the upstream AFBR can translate the message back to (S,G).

When a downstream AFBR receives an E-IP PIM (*,G) message, S' can be generated with the "source address" field set to * (wildcard value). The translated message will be forwarded to the corresponding upstream AFBR. Since every PIM router within a PIM domain MUST be able to map a particular multicast group address to the same RP when the source address is set to wildcard value (see Section 4.7 of [RFC7761]), when the upstream AFBR checks the "source address" field of the message, it finds the IPv4 address of the RP, and ascertains that this is originally a (*,G) message. This is then translated back to the (*,G) message and processed.

6. Control Plane Functions of AFBR

AFBRs are responsible for the following functions:

6.1. E-IP (*,G) and (S,G) State Maintenance

E-IP (*,G) and (S,G) state maintenance for an AFBR is the same as E-IP (*,G) and (S,G) state maintenance for an mAFTR described in Section 7.2 of [RFC8114]

6.2. I-IP (S',G') State Maintenance

It is possible that the I-IP transit core runs another, non-transit, I-IP PIM-SSM instance. Since the translated source address starts with the unique "Well-Known" prefix or the ISP-defined prefix that MUST NOT be used by another service provider, mesh multicast will not influence non-transit PIM-SSM multicast at all. When an AFBR receives an I-IP (S',G') message, it MUST check S'. If S' starts with the unique prefix, then the message is actually a translated

E-IP (S,G) or (*,G) message, and the AFBR translate this message back to a PIMv4 message and process it.

6.3. E-IP (S,G,rpt) State Maintenance

When an AFBR wishes to propagate a Join/Prune(S,G,rpt) [RFC7761] message to an I-IP upstream router, the AFBR MUST operate as specified in Section 6.5 and Section 6.6.

6.4. Inter-AFBR Signaling

Assume that one downstream AFBR has joined an RPT of (*,G) and an SPT of (S,G), and decided to perform an SPT switchover (see Section 4.2.1 of [RFC7761]). According to [RFC7761], it should propagate a Prune(S,G,rpt) message along with the periodical Join(*,G) message upstream towards the RP. However, routers in the I-IP transit core do not process (S,G,rpt) messages since the I-IP transit core is treated as SSM-only. As a result, the downstream AFBR is unable to prune S from this RPT, so it will receive two copies of the same data for (S,G). In order to solve this problem, we introduce a new mechanism for downstream AFBRs to inform upstream AFBRs of pruning any given S from an RPT.

When a downstream AFBR wishes to propagate an (S,G,rpt) message upstream, it SHOULD encapsulate the (S,G,rpt) message, then send the encapsulated unicast message to the corresponding upstream AFBR, which we call "RP'".

When RP' receives this encapsulated message, it MUST decapsulate the message as in the unicast scenario, and retrieve the original (S,G,rpt) message. The incoming interface of this message may be different to the outgoing interface which propagates multicast data to the corresponding downstream AFBR, and there may be other downstream AFBRs that need to receive multicast data of (S,G) from this incoming interface, so RP' should not simply process this message as specified in [RFC7761] on the incoming interface.

To solve this problem, we introduce an "interface agent" to process all the encapsulated (S,G,rpt) messages the upstream AFBR receives. The interface agent's RP' should prune S from the RPT of group G when no downstream AFBR is subscribed to receive multicast data of (S,G) along the RPT.

In this way, we ensure that downstream AFBRs will not miss any multicast data that they need. The cost of this is that multicast data for (S,G) will be duplicated along the RPT received by AFBRs affected by the SPT switch over, if at least one downstream AFBR

exists that has not yet sent Prune(S,G,rpt) messages to the upstream AFBR.

In certain deployment scenarios (e.g. if there is only a single downstream router), the interface agent function is not required.

The mechanism used to achieve this is left to the implementation. The following diagram provides one possible solution for an "interface agent" implementation:

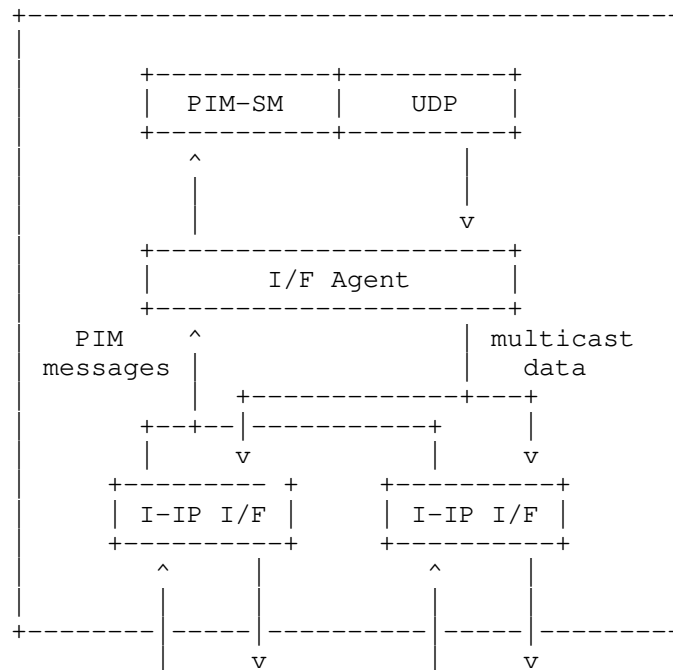


Figure 4: Interface Agent Implementation Example

Figure 4 shows an example of an interface agent implementation using UDP encapsulation. The interface agent has two responsibilities: In the control plane, it should work as a real interface that has joined $(*,G)$, representing of all the I-IP interfaces which are outgoing interfaces of the $(*,G)$ state machine, and process the (S,G,rpt) messages received from all the I-IP interfaces.

The interface agent maintains downstream (S,G,rpt) state machines for every downstream AFBR, and submits Prune (S,G,rpt) messages to the PIM-SM module only when every (S,G,rpt) state machine is in the Prune(P) or PruneTmp(P') state, which means that no downstream AFBR is subscribed to receive multicast data for (S,G) along the RPT of G. Once a (S,G,rpt) state machine changes to NoInfo(NI) state, which means that the corresponding downstream AFBR has switched to receive multicast data of (S,G) along the RPT again, the interface agent MUST send a Join (S,G,rpt) to the PIM-SM module immediately.

In the data plane, upon receiving a multicast data packet, the interface agent MUST encapsulate it at first, then propagate the encapsulated packet from every I-IP interface.

NOTICE: It is possible that an E-IP neighbor of RP' has joined the RPT of G, so the per-interface state machine for receiving E-IP Join/Prune (S,G,rpt) messages should be preserved.

6.5. SPT Switchover

After a new AFBR requests the receipt of traffic destined for a multicast group, it will receive all the data from the RPT at first. At this time, every downstream AFBR will receive multicast data from any source from this RPT, in spite of whether they have switched over to an SPT or not.

To minimize this redundancy, it is recommended that every AFBR's SwitchToSptDesired(S,G) function employs the "switch on first packet" policy. In this way, the delay in switchover to SPT is kept as small as possible, and after the moment that every AFBR has performed the SPT switchover for every S of group G, no data will be forwarded in the RPT of G, thus no more unnecessary duplication will be produced.

6.6. Other PIM Message Types

In addition to Join or Prune, other message types exist, including Register, Register-Stop, Hello and Assert. Register and Register-Stop messages are sent by unicast, while Hello and Assert messages are only used between directly linked routers to negotiate with each other. It is not necessary to translate these for forwarding, thus the processing of these messages is out of scope for this document.

6.7. Other PIM States Maintenance

In addition to states mentioned above, other states exist, including (*,*,RP) and I-IP (*,G') state. Since we treat the I-IP core as SSM-only, the maintenance of these states is out of scope for this document.

7. Data Plane Functions of the AFBR

7.1. Process and Forward Multicast Data

Refer to Section 7.4 of [RFC8114]. If there is at least one outgoing interface whose IP address family is different from the incoming interface, the AFBR MUST encapsulate this packet with mPrefix64-derived and uPrefix64-derived IPv6 address to form an IPv6 multicast packet.

7.2. TTL or Hop Count

Upon encapsulation, the TTL and hop account in the outer header SHOULD be set by policy. Upon decapsulation, the TTL and hop count in the inner header SHOULD be modified by policy, it MUST NOT be incremented and it MAY be decremented to reflect the cost of tunnel forwarding. Besides, processing of TTL and hop count information in protocol headers depends on the tunneling technology, which is out of scope of this document.

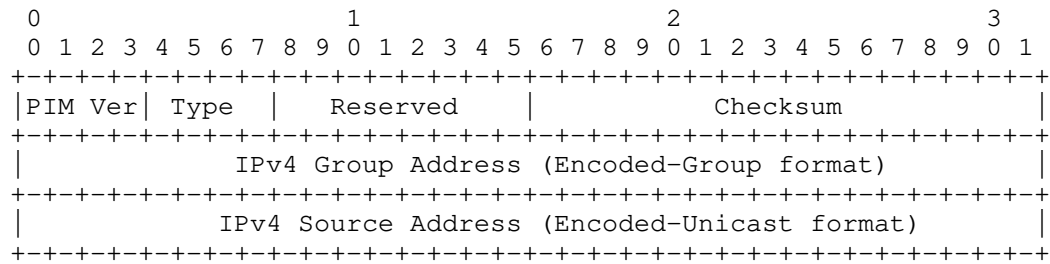
7.3. Fragmentation

The encapsulation performed by an upstream AFBR will increase the size of packets. As a result, the outgoing I-IP link MTU may not accommodate the larger packet size. It is not always possible for core operators to increase the MTU of every link, thus source fragmentation after encapsulation and reassembling of encapsulated packets MUST be supported by AFBRs [RFC5565]. PMTUD [RFC8201] SHOULD be enabled and ICMPv6 packets MUST NOT be filtered in the I-IP network. Fragmentation and tunnel configuration considerations are provided in Section 8 of [RFC5565]. The detailed procedure can be referred in Section 7.2 of [RFC2473].

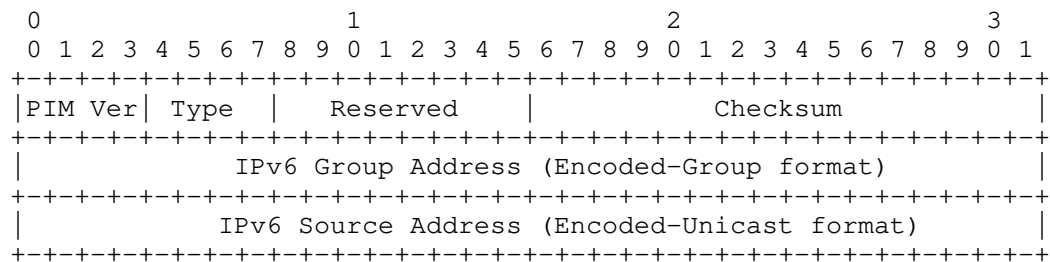
8. Packet Format and Translation

Because the PIM-SM Specification is independent of the underlying unicast routing protocol, the packet format in Section 4.9 of [RFC7761] remains the same, except that the group address and source address MUST be translated when traversing an AFBR.

For example, Figure 5 shows the register-stop message format in the IPv4 and IPv6 address families.



(1). IPv4 Register-Stop Message Format



(2). IPv6 Register-Stop Message Format

Figure 5: Register-Stop Message Format

In Figure 5, the semantics of fields "PIM Ver", "Type", "Reserved", and "Checksum" can be referred in Section 4.9 of [RFC7761].

IPv4 Group Address (Encoded-Group format): The encoded-group format of the IPv4 group address described in Section 4.9.1 of [RFC7761]

IPv4 Source Address (Encoded-Group format): The encoded-unicast format of the IPv4 source address described in Section 4.9.1 of [RFC7761]

IPv6 Group Address (Encoded-Group format): The encoded-group format of the IPv6 group address described in Section 5.2.

IPv6 Source Address (Encoded-Group format): The encoded-unicast format of the IPv6 source address described in Section 5.3.

9. Softwire Mesh Multicast Encapsulation

Softwire mesh multicast encapsulation does not require the use of any one particular encapsulation mechanism. Rather, it MUST accommodate a variety of different encapsulation mechanisms, and allow the use of encapsulation mechanisms mentioned in [RFC4925]. Additionally, all of the AFBRs attached to the I-IP network MUST implement the same

encapsulation mechanism, and follow the requirements mentioned in Section 8 of [RFC5565].

10. Security Considerations

The security concerns raised in [RFC4925] and [RFC7761] are applicable here.

The additional workload associated with some schemes, such as interface agents, could be exploited by an attacker to perform a DDOS attack.

Compared with [RFC4925], the security concerns should be considered more carefully: an attacker could potentially set up many multicast trees in the edge networks, causing too many multicast states in the core network. To defend against these attacks, BGP policies SHOULD be carefully configured, e.g., AFBRS only accept Well-Known prefix advertisements from trusted peers. Besides, cryptographic methods for authenticating BGP sessions [RFC7454] could be used.

11. IANA Considerations

This document includes no request to IANA.

12. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <<https://www.rfc-editor.org/info/rfc2473>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.
- [RFC4925] Li, X., Ed., Dawkins, S., Ed., Ward, D., Ed., and A. Durand, Ed., "Softwire Problem Statement", RFC 4925, DOI 10.17487/RFC4925, July 2007, <<https://www.rfc-editor.org/info/rfc4925>>.
- [RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, DOI 10.17487/RFC5565, June 2009, <<https://www.rfc-editor.org/info/rfc5565>>.

- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, DOI 10.17487/RFC6052, October 2010, <<https://www.rfc-editor.org/info/rfc6052>>.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <<https://www.rfc-editor.org/info/rfc6333>>.
- [RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <<https://www.rfc-editor.org/info/rfc6513>>.
- [RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454, February 2015, <<https://www.rfc-editor.org/info/rfc7454>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC7899] Morin, T., Ed., Litkowski, S., Patel, K., Zhang, Z., Kebler, R., and J. Haas, "Multicast VPN State Damping", RFC 7899, DOI 10.17487/RFC7899, June 2016, <<https://www.rfc-editor.org/info/rfc7899>>.
- [RFC8114] Boucadair, M., Qin, C., Jacquenet, C., Lee, Y., and Q. Wang, "Delivery of IPv4 Multicast Services to IPv4 Clients over an IPv6 Multicast Network", RFC 8114, DOI 10.17487/RFC8114, March 2017, <<https://www.rfc-editor.org/info/rfc8114>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <<https://www.rfc-editor.org/info/rfc8201>>.

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: xumw@tsinghua.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

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

Shu Yang
Shenzhen University
South Campus, Shenzhen University
Shenzhen 518060
P.R. China

Phone: +86-755-2653-4078
Email: yang.shu@szu.edu.cn

Chris Metz
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

Phone: +1-408-525-3275
Email: chmetz@cisco.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: October 1, 2013

N. Matsuhira
Fujitsu Limited
March 30, 2013

SA46T Multicast Support
draft-matsuhira-sa46t-mcast-03

Abstract

This document describe Stateless Automatic IPv4 over IPv6 Encapsulation / Decapsulation Technology (SA46T) multicast support. IPv4 multicast is supported by SA46T with same manner with IPv4 unicast. SA46T multicast address prefix is defined.

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 October 1, 2013.

Copyright Notice

Copyright (c) 2013 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	3
2. Architecture of SA46T Multicast	3
3. Format of SA46T Multicast address	5
4. IANA Considerations	6
5. Security Considerations	6
6. Acknowledgements	6
7. References	7
7.1. Normative References	7
7.2. References	7
Author's Address	7

1. Introduction

This document describe Stateless Automatic IPv4 over IPv6 Encapsulation / Decapsulation Technology (SA46T) multicast support.

SA46T [I-D.draft-matsuhira-sa46t-spec] makes backbone network to IPv6 only. And also, SA46T can stack many IPv4 networks, i.e. the networks using same IPv4 (private) address, without interdependence.

IPv4 multicast is supported by SA46T with same manner with IPv4 unicast.

2. Architecture of SA46T Multicast

IPv4 multicast address is known as Class D IPv4 address, 224.0.0.0/4. The range is from 224.0.0.0 to 239.255.255.255.

Mapping IPv4 multicast address to IPv6 addressing space, the IPv6 address which mapped to IPv4 mapped address is also IPv6 multicast address space, because copy of the packet for multicasting may occur not only in IPv4 subnet but also in IPv6 backbone. So, SA46T multicast support requires special IPv6 address prefix, SA46T multicast address prefix.

Figure 1 shows address architecture of SA46T and SA46T multicast support. Both unicast case and multicast case, mapping IPv4 address is the same, and usage of IPv4 network plane ID is the same, however, SA46T address prefix (unicast) and SA46T multicast address prefix is not the same. If value of IPv4 network plane ID is the same, IPv4 unicast address and IPv4 multicast address belong to the same network plane.

In this document, SA46T unicast address which contain IPv4 unicast address, and SA46T multicast address which contain IPv4 multicast address are used.

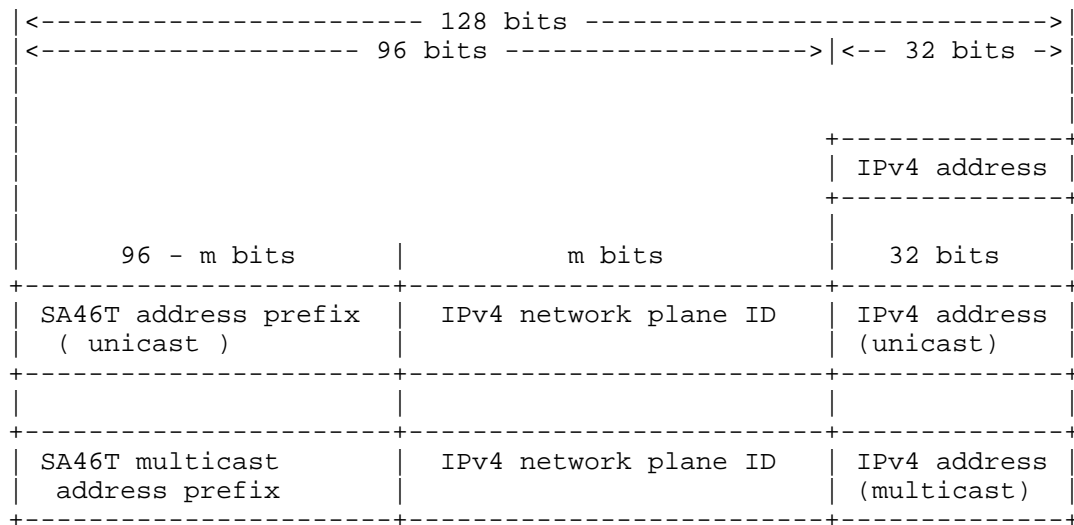


Figure 1

Figure 2 show the format of SA46T multicast address prefix. IPv6 multicast address has a prefix of FF00::/8. SA46T multicast address prefix should be the same with IPv6 multicast address prefix.

A group ID part of IPv6 multicast address is mapped with reserve space, IPv4 network plane ID, and IPv4 (multicast) address.

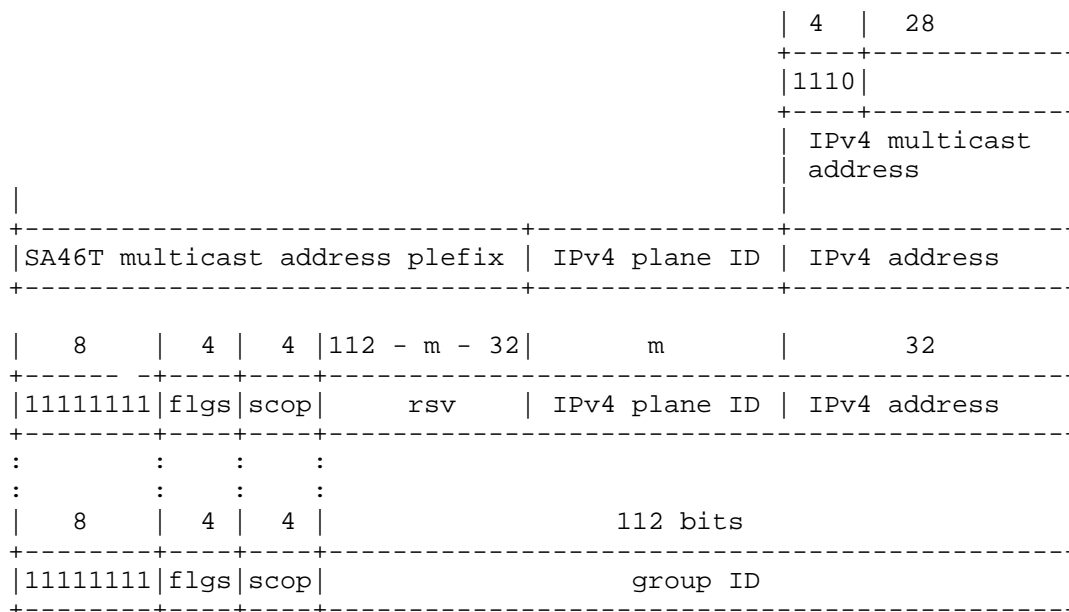


Figure 2

The information which indicate SA46T multicast address is needed, because there is a possibility the value of group ID for IPv6 multicast and the value of "rsv+IPv4 network plane ID + IPv4 multicast address" is the same.

This information is TBD.

However, this version of this document suppose using flag space. The flag space consists four flags (bits), and high-order 3 flags are reserved, and 4th flag is T, which indicate the address is permanently-assigned (well-known) or non-permanently-assigned (transient).

There is a idea, which allocate SA46T multicast address flag in flag space, and using scop space also other idea, too

3. Format of SA46T Multicast address

This example is based on IPv6 Global Unicast Address Format [RFC3587].

Figure 3 shows IPv6 Global Unicast Address Format.

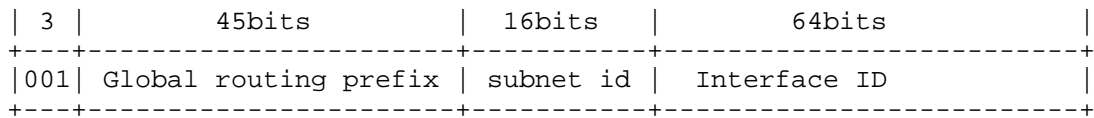


Figure 3

Figure 4 shows SA46T multicast address format using part of IPv6 Global Unicast Address.

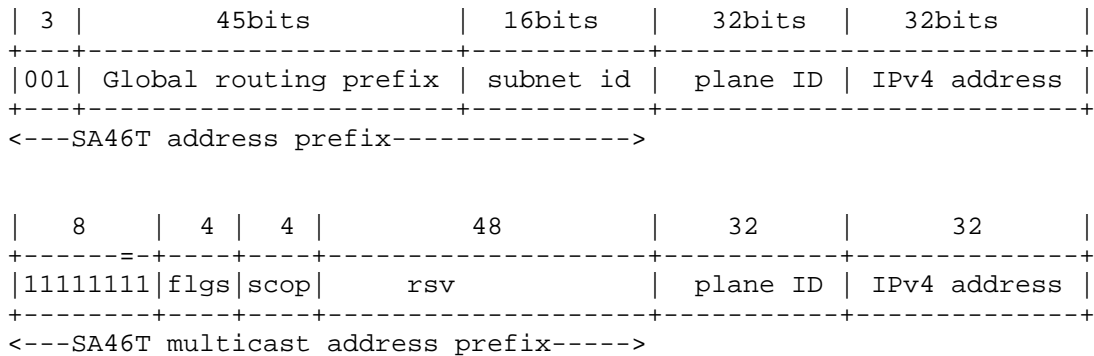


Figure 4

4. IANA Considerations

This document may make request of IANA.

5. Security Considerations

SA46T use automatic encapsulation technologies. Security consideration related tunneling technologies are discussed in RFC2893[RFC2893], RFC2267[RFC2267], etc.

6. Acknowledgements

7. References

7.1. Normative References

- [I-D.draft-matsuhira-sa46t-spec]
Matsuhira, N., "Stateless Automatic IPv4 over IPv6
Encapsulation / Decapsulation Technology: Specification",
January 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3587] Hinden, R., Deering, S., and E. Nordmark, "IPv6 Global
Unicast Address Format", RFC 3587, August 2003.

7.2. References

- [RFC2267] Ferguson, P. and D. Senie, "Network Ingress Filtering:
Defeating Denial of Service Attacks which employ IP Source
Address Spoofing", RFC 2267, January 1998.
- [RFC2893] Gilligan, R. and E. Nordmark, "Transition Mechanisms for
IPv6 Hosts and Routers", RFC 2893, August 2000.

Author's Address

Naoki Matsuhira
Fujitsu Limited
1-1, Kamikodanaka 4-chome, Nakahara-ku
Kawasaki, 211-8588
Japan

Phone: +81-44-754-3466
Fax:
Email: matsuhira@jp.fujitsu.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: May 3, 2012

M. Mawatari
Japan Internet Exchange Co.,Ltd.
M. Kawashima
NEC AccessTechnica, Ltd.
C. Byrne
T-Mobile USA
October 31, 2011

464XLAT: Combination of Stateful and Stateless Translation
draft-mawatari-softwire-464xlat-02

Abstract

This document describes a method (464XLAT) for IPv4 connectivity across IPv6 network by combination of stateful translation and stateless translation. 464XLAT is a simple technique to provide IPv4 access service while avoiding encapsulation by using twice IPv4/IPv6 translation standardized in [RFC6145] and [RFC6146].

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 May 3, 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	3
2. Requirements Language	3
3. Terminology	3
4. Network Architecture	4
4.1. Wireline Network Architecture	4
4.2. Wireless 3GPP Network Architecture	5
5. Applicability	5
5.1. Wireline Network Applicability	5
5.2. Wireless 3GPP Network Applicability	6
6. Implementation Considerations	6
6.1. IPv6 Address Format	6
6.2. DNS Proxy Implementation	7
6.3. IPv6 Fragment Header Consideration	7
6.4. Auto Prefix Assignment	7
7. Deployment Considerations	7
8. Security Considerations	8
9. IANA Considerations	8
10. Acknowledgements	8
11. References	9
11.1. Normative References	9
11.2. Informative References	9
Authors' Addresses	10

1. Introduction

The IANA unallocated IPv4 address pool was exhausted on February 3, 2011. It is likely that each RIR's unallocated IPv4 address pool will exhaust in the near future. In this situation, it will be difficult for many networks to assign IPv4 address to end users despite substantial IPv4 connectivity required for mobile devices, smart-grid, and cloud nodes.

This document describes an IPv4 over IPv6 solution as one of the measures of IPv4 address extension and encouragement of IPv6 deployment.

The 464XLAT method described in this document uses twice IPv4/IPv6 translation standardized in [RFC6145] and [RFC6146]. It does not require DNS64 [RFC6147], but it may use DNS64. It is also possible to provide single IPv4/IPv6 translation service, which will be needed in the near future. This feature is one of the advantages, because it can be an encouragement to gradually transition to IPv6.

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

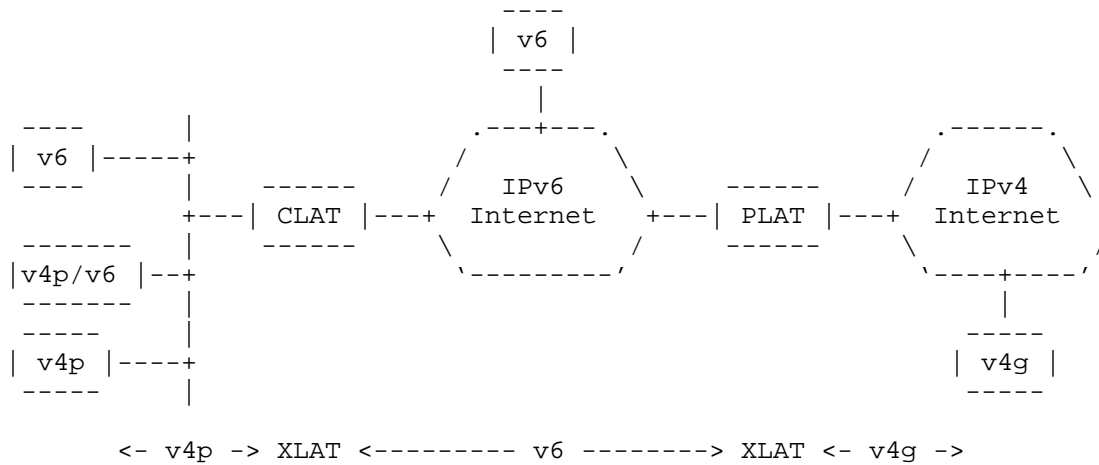
PLAT: PLAT is Provider side translator(XLAT). A stateful translator complies with [RFC6146] that performs 1:N translation. It translates global IPv6 address to global IPv4 address, and vice versa.

CLAT: CLAT is Customer side translator(XLAT). A stateless translator complies with [RFC6145] that performs 1:1 translation. It algorithmically translates private IPv4 address to global IPv6 address, and vice versa. It has also IPv6 router function that can forward IPv6 packet for IPv6 hosts in end-user network. Furthermore, it has DNS Proxy function with IPv6 transport that provides name resolution for IPv4 hosts and IPv6 hosts in end-user network. The presence of DNS64 [RFC6147] and any port mapping algorithm are not required.

4. Network Architecture

464XLAT method is shown in the following figure.

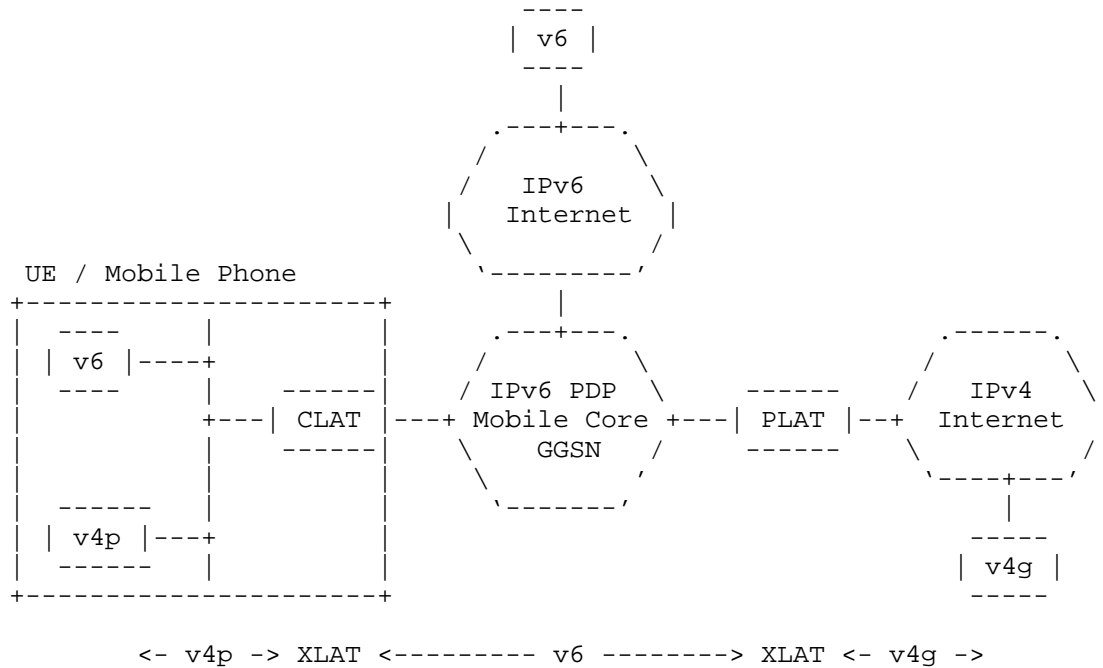
4.1. Wireline Network Architecture



v6 : Global IPv6
 v4p : Private IPv4
 v4g : Global IPv4

Figure 1: Wireline Network Topology

4.2. Wireless 3GPP Network Architecture



v6 : Global IPv6
v4p : Private IPv4
v4g : Global IPv4

Figure 2: Wireless 3GPP Network Topology

5. Applicability

5.1. Wireline Network Applicability

When ISP has IPv6 access network infrastructure and 464XLAT, ISP can provide IPv4 service to end users.

If the IXP or another provider operates the PLAT, all ISPs have to do is to deploy IPv6 access network. All ISPs do not need IPv4 facilities. They can migrate quickly their operation to an IPv6-only environment. Incidentally, Japan Internet Exchange(JPIX) is providing 464XLAT trial service since July 2010.

5.2. Wireless 3GPP Network Applicability

In pre-release 9 3GPP networks, GSM and UMTS networks must signal and support both IPv4 and IPv6 PDP attachments to access IPv4 and IPv6 network destinations. This is generally not operationally viable since much of the network cost is derived from the number of PDP attachments, both in terms of licenses from the network hardware vendors and in terms of actual hardware resources required to support and maintain the PDP signaling and mobility events. This has been one of the operational challenges of bringing IPv6 to mobile networks, it simply costs more from the network provider perspective and does not result in any new revenues, since customers are not willing to pay for IPv6 access.

Now that both global and private IPv4 addresses are scarce to the extent that it is a substantial business risk and limiting growth in many areas, the mobile network providers must support IPv6 address which solve the IP address scarcity issue, but it is not feasible to simply turn on additional IPv6 PDP network attachments since that does not solve the near-term IPv4 scarcity issues and at it also increases cost. The most logical path forward is to replace IPv6 with IPv4 and replace the common NAT44 with NAT64 and DNS64. Extensive live network testing with hundreds of friendly-users has shown that IPv6-only network attachments for mobile devices covers over 90% of the common use-cases in Symbian and Android mobile operating systems. The remaining 10% of use-cases do not work because the application requires an IPv4 socket or the application references an IPv4-literal.

464XLAT in combination with NAT64 and DNS64 allows 90% of the applications to continue to work with single translation while at the sametime facilitating legacy IPv4-only applications by providing a private IPv4 address and IPv4 route on the host for the applications to reference and bind to. Traffic sourced from the IPv4 interface is immediately routed the NAT46 CLAT function and passed to the IPv6-only mobile network and destined to the PLAT NAT64.

6. Implementation Considerations

6.1. IPv6 Address Format

IPv6 address format in 464XLAT is presented in the following format.

XLAT prefix(96)	IPv4(32)
-----------------	----------

IPv6 Address Format for 464XLAT

Source address and destination address have IPv4 address embedded in the low-order 32 bits of the IPv6 address. The format is defined in Section 2.2 of [RFC6052]. However, 464XLAT does not use the Well-Known Prefix "64:ff9b::/96".

6.2. DNS Proxy Implementation

CLAT perform DNS Proxy for IPv4 hosts and IPv6 hosts in end-user network. It MUST provide name resolution with IPv6 transport. It does not need DNS64 [RFC6147] function.

6.3. IPv6 Fragment Header Consideration

In the 464XLAT environment, the PLAT and CLAT SHOULD include an IPv6 Fragment Header, since IPv4 host does not set the DF bit. However, the IPv6 Fragment Header has been shown to cause operational difficulties in practice due to limited firewall fragmentation support, etc. Therefore, the PLAT and CLAT may provide a configuration function that allows the PLAT and CLAT not to include the Fragment Header for the non-fragmented IPv6 packets. At any rate, both behaviors SHOULD match.

6.4. Auto Prefix Assignment

Source IPv6 prefix assignment in CLAT is via DHCPv6 prefix delegation or another method. Destination IPv6 prefix assignment in CLAT is via some method. (e.g., DHCPv6 option, TR-069, DNS, HTTP, [I-D.ietf-behave-nat64-discovery-heuristic], etc.)

7. Deployment Considerations

Even if the Internet access provider for consumers is different from the PLAT provider (another Internet access provider or Internet exchange provider, etc.), it can implement traffic engineering independently from the PLAT provider. Detailed reasons are below.

1. The Internet access provider for consumers can figure out IPv4 source address and IPv4 destination address from translated IPv6 packet header, so it can implement traffic engineering based on IPv4 source address and IPv4 destination address (e.g. traffic monitoring for each IPv4 destination address, packet filtering

for each IPv4 destination address, etc.). The Tunneling methods do not have such a advantage, without any deep packet inspection for visualizing the inner IPv4 packet of the tunnel packet.

2. If the Internet access provider for consumers can assign IPv6 prefix greater than /64 for each subscriber, this 464XLAT method can separate IPv6 prefix for native IPv6 packets and XLAT prefix for IPv4/IPv6 translation packets. Accordingly, it can identify the type of packets ("native IPv6 packets" and "IPv4/IPv6 translation packets"), and implement traffic engineering based on IPv6 prefix.

This 464XLAT method have two capabilities. One is a IPv6 -> IPv4 -> IPv6 translation for sharing global IPv4 addresses, another is a IPv4 -> IPv6 translation for reaching IPv6 only servers from IPv4 only clients that can not support IPv6. IPv4 only clients will remain for a while.

8. Security Considerations

To implement a PLAT, see security considerations presented in Section 5 of [RFC6146].

To implement a CLAT, see security considerations presented in Section 7 of [RFC6145]. And furthermore, the CLAT SHOULD perform Bogon filter, and SHOULD have IPv6 firewall function as a IPv6 router. It is useful function for native IPv6 packet and translated IPv6 packet. The CLAT SHOULD check IPv6 packet received from WAN interface. If the packet is invalid prefix (i.e., it is not XLAT prefix), then SHOULD silently drop the packet. In addition, the CLAT SHOULD check IPv4 packet after the translation. If the packet is not match private IPv4 address of LAN, then SHOULD silently drop the packet.

9. IANA Considerations

This document has no actions for IANA.

10. Acknowledgements

The authors would like to thank JPIX NOC members and Seiichi Kawamura for their helpful comments.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

11.2. Informative References

- [I-D.ietf-behave-nat64-discovery-heuristic] Savolainen, T. and J. Korhonen, "Discovery of a Network-Specific NAT64 Prefix using a Well-Known Name", draft-ietf-behave-nat64-discovery-heuristic-03 (work in progress), October 2011.
- [I-D.ietf-v6ops-3gpp-eps] Korhonen, J., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3GPP Evolved Packet System", draft-ietf-v6ops-3gpp-eps-08 (work in progress), September 2011.
- [I-D.murakami-softwire-4v6-translation] Murakami, T., Chen, G., Deng, H., Dec, W., and S. Matsushima, "4via6 Stateless Translation", draft-murakami-softwire-4v6-translation-00 (work in progress), July 2011.
- [I-D.xli-behave-divi] Bao, C., Li, X., Zhai, Y., and W. Shang, "dIVI: Dual-Stateless IPv4/IPv6 Translation", draft-xli-behave-divi-04 (work in progress), October 2011.
- [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

Authors' Addresses

Masataka Mawatari
Japan Internet Exchange Co.,Ltd.
KDDI Otemachi Building 19F, 1-8-1 Otemachi,
Chiyoda-ku, Tokyo 100-0004
JAPAN

Phone: +81 3 3243 9579
Email: mawatari@jpix.ad.jp

Masanobu Kawashima
NEC AccessTechnica, Ltd.
800, Shimomata
Kakegawa-shi, Shizuoka 436-8501
JAPAN

Phone: +81 537 23 9655
Email: kawashimam@vx.jp.nec.com

Cameron Byrne
T-Mobile USA
Bellevue, Washington 98105
USA

Email: cameron.byrne@t-mobile.com

Softwire WG
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2013

T. Mrugalski
ISC
O. Troan
Cisco
C. Bao
Tsinghua University
W. Dec
Cisco
July 4, 2012

DHCPv6 Options for Mapping of Address and Port
draft-mdt-softwire-map-dhcp-option-03

Abstract

This document specifies DHCPv6 options for the provisioning of Mapping of Address and Port (MAP) Customer Edge (CE) devices, based on the MAP parameters defined in [I-D.ietf-softwire-map].

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, 2013.

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. Introduction	3
2. Conventions	3
3. MAP Information	4
4. DHCPv6 MAP Options Format	4
4.1. MAP Option	5
4.2. MAP Rule Option	6
4.3. MAP DMR Option	8
4.4. MAP Port Parameters Option	8
5. MAP Options Examples	9
5.1. BMR Option Example	9
5.2. FMR Option Example	10
5.3. DMR Option Example	10
6. DHCPv6 Server Behavior	10
7. DHCPv6 Client Behavior	10
8. Usage of flags and paramaters	11
9. Deployment considerations	12
10. IANA Considerations	12
11. Security Considerations	13
12. Acknowledgements	13
13. References	13
13.1. Normative References	13
13.2. Informative References	14
Authors' Addresses	15

1. Introduction

Mapping of Address and Port (MAP) defined in [I-D.ietf-softwire-map] is a mechanism for providing IPv4 connectivity service to end users over a service provider's IPv6 network, allowing for shared or dedicated IPv4 addressing. It consists of a set of one or more MAP Border Relay (BR) routers, responsible for stateless forwarding, and one or more MAP Customer Edge (CE) routers, that collectively form a MAP Domain when configured with common MAP rule-sets. In a residential broadband deployment the CE is sometimes referred to as a Residential Gateway (RG) or Customer Premises Equipment (CPE).

A typical MAP CE will serve its end-user with one WAN side interface connected to an operator domain providing a MAP service. To function in the MAP domain, the CE requires to be provisioned with the appropriate MAP service parameters for that domain. Particularly in larger networks it is not feasible to configure such parameters manually, which forms the requirement for a dynamic MAP provisioning mechanism that is defined in this document based on the existing DHCPv6 [RFC3315] protocol. The configuration of the MAP BR is outside of scope of this document.

This document specifies the DHCPv6 options that allow MAP CE provisioning, based on the definitions of parameters provided in [I-D.ietf-softwire-map], and is applicable to both MAP-E and MAP-T transport variants. The definition of DHCPv6 options for MAP CE provisioning does not preclude the definition of other dynamic methods for configuring MAP devices, or supplementing such configuration, nor is the use of DHCPv6 provisioning mandatory for MAP operation.

Since specification of MAP architecture is still expected to evolve, DHCPv6 options may have to evolve too to fit the revised MAP specification.

Described proposal is not a dynamic port allocation mechanism.

Readers interested in deployment considerations are encouraged to read [I-D.mdt-softwire-map-deployment].

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 RFC 2119 [RFC2119].

3. MAP Information

The following presents the information parameters that are used to configure a MAP CE:

- o A Default Mapping Rule (DMR). This rule governs the default forwarding/mapping behaviour of the MAP CE, ie it informs the CE of the BR router's address or prefix that is typically used as a default. The DMR is a mandatory parameter for a MAP CE.
- o A Basic Mapping Rule (BMR). This rule governs the MAP configuration of the CE, including that of completing the CE's MAP IPv6 address, as well as deriving the CE's IPv4 parameters. Key parameters of a BMR include: i) The IPv4 Prefix - Used to derive the CE's IPv4 address; ii) The Embedded Address bit length - Used to derive how many, if any, of the CE's IPv6 address is mapped to the IPv4 address. iii) The IPv6 prefix - used to determine the CE's IPv6 MAP domain prefix that is to form the base for the CE's MAP address. The BMR is an optional rule for a MAP CE.
- o A Forward Mapping Rule (FMR). This rule governs the MAP CE-CE forwarding behaviour for IPv4 destinations covered by the rule. The FMR is effectively a special type of an BMR, given that it shares exactly the same configuration parameters, except that these parameters are only applied for setting up forwarding. Its presence enables a given CE to communicate directly in "mesh mode" with other CEs. The FMR is an optional rule, and the absence of such a rule indicates that the CE is to simply use its default mapping rule for all destinations.
- o Transport mode; encapsulation (MAP-E) or translation (MAP-T) modes to be used for the MAP CE Domain.
- o Additional parameters. The MAP specification allows great flexibility in the level of automation a CE uses to derive its IPv4 address and port-sharing (PSID), ranging from full derivation of these parameters from the CE's IPv6 prefix, to full parametrization of MAP configuration independent of the CE's IPv6 prefix. Optional parameters such as the PSID allow this flexibility.

4. DHCPv6 MAP Options Format

The DHCPv6 protocol is used for MAP CE provisioning following regular DHCPv6 notions, with the MAP CE assuming a DHCPv6 client role, and the MAP parameters provided by the DHCPv6 server following server side policies. The format and usage of the MAP options is defined in the following sections.

Discussion: As the exact parameters required to configure MAP rules and MAP in general are expected to change, this section is expected

to be updated and follow change in the [I-D.ietf-softwire-map].

Discussion: It should be noted that initial concept of 4rd/MAP provisioning was presented in DHC working group meeting. It used one complex option to convey all required parameters. Strong suggestion from DHC WG was to use several simpler options. Options (possibly nested) are preferred over conditional option formatting. See DHCP option guidelines document [I-D.ietf-dhc-option-guidelines]).

Server that supports MAP configuration and is configured to provision requesting CE MUST include exactly one OPTION_MAP option in a REPLY message for each MAP domain. It is envisaged that in typical network, there will be only one MAP domain deployed.

4.1. MAP Option

This MAP Option specifies the container option used to group all rules for a specified MAP domain.

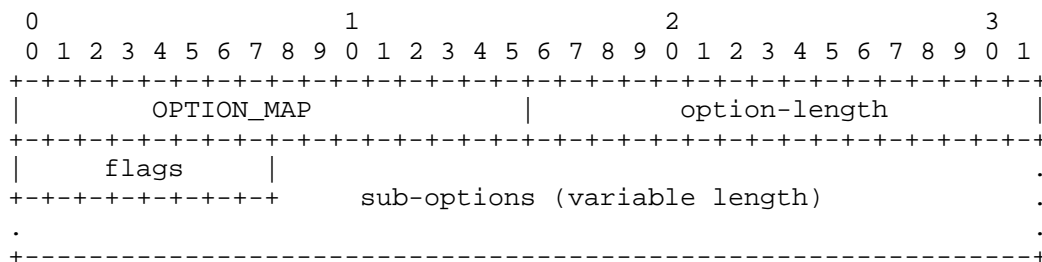


Figure 1: MAP Option

- o option-code: OPTION_MAP (TBD1)
- o option-length: 1 + Length of the sub-options
- o flags: This 8-bits long conveys the MAP Option Flags. The meaning of specific bits is explained in Figure 2.
- o sub-options: options associated to this MAP option.

The sub options field encapsulates those options that are specific to this MAP Option. For example, all of the MAP Rule Options are in the sub-options field. A DHCP message may contain multiple MAP Options.

The Format of the MAP Option Flags field is:

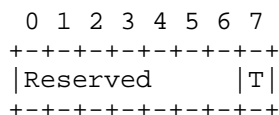


Figure 2: MAP Option Flags

- o Reserved: 7-bits reserved for future use.
- o T: 1 bit field that specifies transport mode to use: translation (0) or encapsulation (1).

It was suggested to also provision information whether MAP network is working in hub and spoke or mesh mode. That is not necessary, as mesh mode is assumed when there is at least one FMR present.

4.2. MAP Rule Option

Figure X shows the format of the MAP Rule option used for conveying the BMR and FMR.

Server includes one or more MAP Rule Options in MAP Flags option.

Server MAY send more than one MAP Rule Option, if it is configured to do so. Clients MUST NOT send MAP Rule Option.

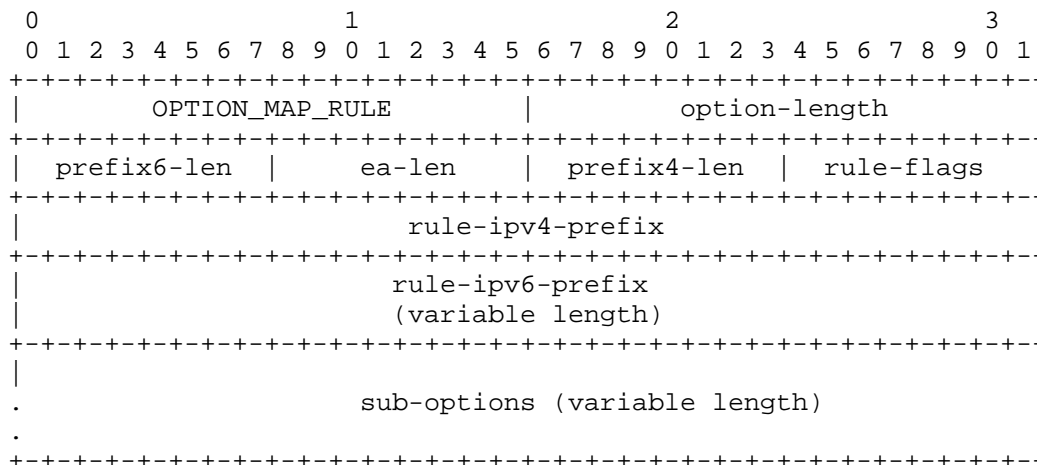


Figure 3: MAP Rule Option

- o option-code: OPTION_MAP_RULE (TBD2)

- o option-length: length of the option, excluding option-code and option-length fields, including length of all sub-options.
- o prefix6-len: 8 bits long field expressing the bit mask length of the IPv6 prefix specified in the rule-ipv6-prefix field.
- o ea-len: 8-bits long field that specifies the Embedded-Address (EA) bit length. Values allowed range from 0 to 48.
- o prefix4-len: 8 bits long field expressing the bit mask length of the IPv4 prefix specified in the rule-ipv4-prefix field.
- o rule-flags: 8 bit long field carrying flags applicable to the rule. The meaning of specific bits is explained in Figure 4.
- o rule-ipv4-prefix: a 32 bit fixed length field that specifies the IPv4 prefix for the MAP rule.
- o rule-ipv6-prefix: a variable length field that specifies the IPv6 domain prefix for the MAP rule. The field is padded with zeros up to the nearest octet boundary when prefix6-len is not divisible by 8.
- o rule sub-options: a variable field that may contain zero or more options that specify additional parameters for this MAP BMR/FMR rule. Currently there is only one option defined that may appear in rule sub-options field, eg the OPTION_MAP_PORTPARAMS, defined in section Section 4.4.

The value of the EA-len and prefix4-len SHOULD be equal to or greater than 32.

The Format of the MAP Rule Flags field is:

```

      0 1 2 3 4 5 6 7
      +---+---+---+---+
      |Reserved       |F|
      +---+---+---+---+

```

Figure 4: MAP Rule Flags

- o Reserved: 7-bits reserved for future use as flags.
- o F-Flag: 1 bit field that specifies whether the rule is to be used for forwarding (FMR). 0x0 = This rule is NOT used as an FMR. 0x1 = This rule is also an FMR.
- o Note: BMR rules can be also FMR rules by setting the F flag. BMR rules are determined by a match of the Rule-IPv6-prefix against the CPE's prefix(es).

It is expected that in a typical MAP deployment scenarios, there will be a single DMR and a single BMR, which could also be designated as an FMR using the F-Flag.

4.3. MAP DMR Option

Figure X shows the format of the MAP Rule option used for conveying the DMR.

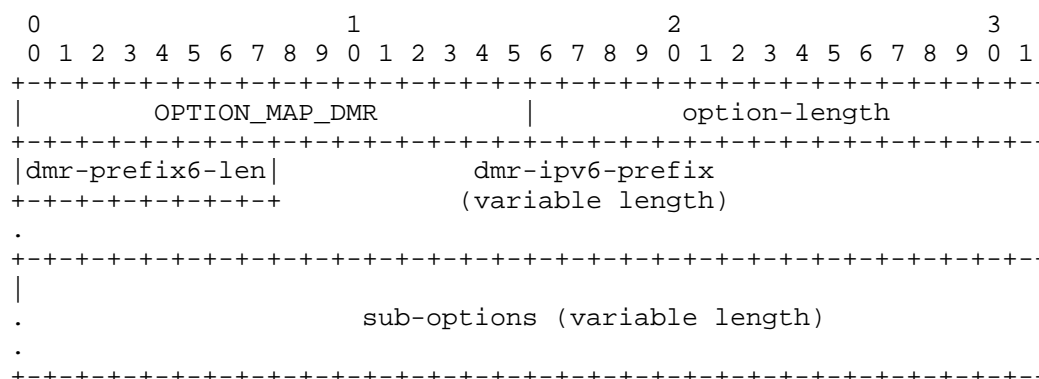


Figure 5: MAP DMR Option

- o option-code: OPTION_MAP_DMR (TBD3)
- o option-length: 1 + length of dmr-ipv6-prefix + sub-options in bytes
- o dmr-prefix6-len: T8 bits long field expressing the bit mask length of the IPv6 prefix specified in the dmr-ipv6-prefix field.
- o dmr-ipv6-prefix: a variable length field that specifies the IPv6 prefix or address for the MAP BR. This field is padded with zeros up to the nearest octet boundary when prefix4-len is not divisible by 8.
- o sub options: options associatied to this MAP DMR option.

4.4. MAP Port Parameters Option

Port Parameters Option specifies optional Rule Port Parameters that MAY be provided as part of the Mapping Rule. It MAY appear as sub-option in OPTION_MAP_RULE option. It MUST NOT appear directly in a message.

See [I-D.ietf-softwire-map], Section 5.1 for detailed description of Port mapping algorithm.

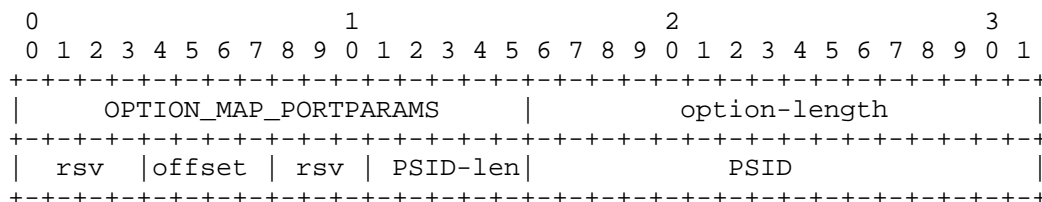


Figure 6: MAP Port Parameters Option

- o option-code: OPTION_MAP_PORTPARAMS (TBD4)
- o option-length: 4
- o rsvd: This 4-bits long field is currently not used and MUST be set to 0 by server. Its value MUST be ignored by clients.
- o offset: (PSID offset) 4 bits long field that specifies the numeric value for the MAP algorithm's excluded port range/offset bits (A-bits), as per section 5.1.1 in [I-D.ietf-softwire-map]. Default must be set to 4.
- o PSID-len: Bit length value of the number of significant bits in the PSID field. (also known as 'k'). When set to 0, the PSID field is to be ignored. After the first 'a' bits, there are k bits in the port number representing valid of PSID. Subsequently, the address sharing ratio would be 2^k .
- o PSID: Explicit 16-bit (unsigned word) PSID value. The PSID value algorithmically identifies a set of ports assigned to a CE. The first k-bits on the left of this 2-octets field is the PSID value. The remaining (16-k) bits on the right are padding zeros.

When receiveing the Port Parameters option with an explicit PSID, the client MUST use this explicit PSID in configuring its MAP interface.

5. MAP Options Examples

DHCPv6 server provisioning a single MAP Rule to a CE (DHCPv6 client) will convey the following MAP options in its messages:

5.1. BMR Option Example

TODO: Reflect example in section 5.2 of MAP draft

Figure 7: BMR Option Example

5.2. FMR Option Example

TODO: Reflect example in section 5.3 of MAP draft

Figure 8: FMR Option Example

5.3. DMR Option Example

TODO: Reflect example in section 5.4 of MAP draft

Figure 9: DMR Option Examples

6. DHCPv6 Server Behavior

RFC 3315 Section 17.2.2 [RFC3315] describes how a DHCPv6 client and server negotiate configuration values using the ORO. As a convenience to the reader, we mention here that a server will by default not reply with a MAP Rule Option if the client has not explicitly enumerated it on its Option Request Option.

A Server following this specification MUST allow the configuration of one or more MAP Rule Options, and SHOULD send such options grouped under a single MAP_OPTION.

Server MUST transmit all configured instances of the Mapping Rule Options with all sub-options, if client requested it using OPTION_MAP_RULE in its Option Request Option (ORO). Server MUST transmit MAP Flags Option if client requested OPTION_MAP in its ORO.

The server MUST be capable of following per client assignment rules when assigning MAP options.

7. DHCPv6 Client Behavior

A MAP CE acting as DHCPv6 client will request MAP configuration to be assigned by the DHCPv6 server located in the ISP network. A client supporting MAP functionality SHOULD request OPTION_MAP, OPTION_MAP_RULE and OPTION_MAP_DMR options in its ORO in SOLICIT, REQUEST, RENEW, REBIND and INFORMATION-REQUEST messages.

When processing received MAP options the following behaviour is expected:

- o A client MUST support processing multiple received OPTION_MAP_RULE options in a OPTION_MAP option
- o A client receiving an unsupported MAP option, or an unrecognized parameter value SHOULD discard the entire OPTION_MAP.
- o Only one OPTION_MAP_DMR is allowed per OPTION_MAP option.

The client MUST be capable of applying the received MAP option parameters for the configuration of the local MAP instance.

Note that system implementing MAP CE functionality may have multiple network interfaces, and these interfaces may be configured differently; some may be connected to networks that call for MAP, and some may be connected to networks that are using normal dual stack or other means. The MAP CE system should approach this specification on an interface-by-interface basis. For example, if the CE system is attached to multiple networks that provide the MAP Mapping Rule Option, then the CE system MUST configure a MAP connection (i.e. a translation or encapsulation) for each interface separately as each MAP provides IPv4 connectivity for each distinct interface. Means to bind a MAP configuration to a given interface in a multiple interfaces device are out of scope of this document.

8. Usage of flags and paramaters

The defined MAP options contain a number of flags and parameters that are intended to provide full flexibility in the configuration of a MAP CE. Some usage examples are:

- o A MAP CE receiving an OPTION_MAP option with the T flag set to 1 will assume a MAP-E (encapsulation) mode of operation for the domain and all associated rules. Conversely, when the received option has the T flag set to 0, the CE will assume a MAP-T (stateless NAT46 translation) mode of operation.
- o The presence of a OPTION_MAP_RULE option, along with IPv4 prefix parameters, indicates to the MAP CE that NAPT44 mode of operation is expected, following the address mapping rules defined in [I-D.ietf-software-map]. Conversely, the absence of an OPTION_MAP_RULE option indicates that NAT44 mode is not required, and that the MAP CE is to plainly encapsulate (MAP-E mode) or statelessly translate using NAT64 (MAP-T mode) any IPv4 traffic sent following the DMR.
- o The MAP domain ipv6-prefix in the BMR should correspond to a service prefix assigned to the CPE by the operator, with the latter being assigned using regular IPv6 means, eg DHCP PD or SLAAC. This parameter allows the CPE to select the prefix for MAP operation.

- o The EA_LEN parameter, along with the length of the IPv4 prefix in the BMR option, allows the MAP CE to determine whether address sharing is in effect, and what is the address sharing ratio. Eg: A prefix4-len of 16 bits, and EA-len of 18 combines to a 32 bit IPv4 address with a sharing ratio of 4.
- o The use of the F(orward) flag in the BMR allows a CE to apply a received BMR as an FMR, thereby enabling mesh-mode for the domain covered by the BMR rule.
- o In the absence of a BMR, the presence of the mandatory DMR indicates to the CPE the address or prefix of a BR, and makes the MAP CE fully compatible with DS-Lite and stateful or stateless NAT64 core nodes. Eg a MAP CE configured in MAP-E mode, with just a DMR and a BR IPv6 address equivalent to that of the AFTR, effectively acts as a DS-Lite B4 element. For more discussion about MAP deployment considerations, see [I-D.mdt-software-map-deployment].

9. Deployment considerations

Usage of PSID Option should be avoided if possible and PSID embedded in the delegated prefix should be used instead. This allows MAP deployment to not introduce any additional state in DHCP server. PSID Option must be assigned on a per CE basis, thus requiring more complicated server configuration.

In a typical environment, there will be only one MAP domain, so server will provide only a single instance of MAP option that acts a container for MAP Rule Options and other options that are specific to that MAP domain.

In case of multiple provisioning domains, as defined in [I-D.ietf-homenet-arch], one server may be required to provide information about more than one MAP domain. In such case, server will provide two or more instances of MAP Options, each with its own set of sub-option that define MAP rules for each specific MAP domain. Details of multiple provisioning domains are discussed in Section 4.1 of [I-D.mdt-software-map-deployment].

10. IANA Considerations

IANA is kindly requested to allocate DHCPv6 option codes for TBD1 for OPTION_MAP, TBD2 for OPTION_MAP_RULE, TBD3 for OPTION_MAP_DMR, and TBD4 for OPTION_MAP_Port. All values should be added to the DHCPv6 option code space defined in Section 24.3 of [RFC3315].

11. Security Considerations

Implementation of this document does not present any new security issues, but as with all DHCPv6-derived configuration state, it is completely possible that the configuration is being delivered by a third party (Man In The Middle). As such, there is no basis to trust that the access over the MAP can be trusted, and it should not therefore bypass any security mechanisms such as IP firewalls.

Readers concerned with security of MAP provisioning over DHCPv6 are encouraged to familiarize with [I-D.ietf-dhc-secure-dhcpv6].

Section XX of [I-D.ietf-softwire-map] discusses security issues of the MAP mechanism.

Section 23 of [RFC3315] discusses DHCPv6-related security issues.

12. Acknowledgements

This document was created as a product of a MAP design team. Following people were members of that team: Congxiao Bao, Mohamed Boucadair, Gang Chen, Maoke Chen, Wojciech Dec, Xiaohong Deng, Jouni Korhonen, Xing Li, Satoru Matsushima, Tomasz Mrugalski, Tetsuya Murakami, Jacni Qin, Necj Scoberne, Qiong Sun, Tina Tsou, Dan Wing, Leaf Yeh and Jan Zorz.

Former MAP design team members are: Remi Despres.

13. References

13.1. Normative References

- [I-D.ietf-softwire-map]
Troan, O., Dec, W., Li, X., Bao, C., Zhai, Y., Matsushima, S., and T. Murakami, "Mapping of Address and Port (MAP)", draft-ietf-softwire-map-01 (work in progress), June 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633,

December 2003.

13.2. 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.ietf-dhc-option-guidelines]
Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and
S. Krishnan, "Guidelines for Creating New DHCPv6 Options",
draft-ietf-dhc-option-guidelines-08 (work in progress),
June 2012.
- [I-D.ietf-dhc-secure-dhcpv6]
Jiang, S. and S. Shen, "Secure DHCPv6 Using CGAs",
draft-ietf-dhc-secure-dhcpv6-06 (work in progress),
March 2012.
- [I-D.ietf-homenet-arch]
Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil,
"Home Networking Architecture for IPv6",
draft-ietf-homenet-arch-03 (work in progress), June 2012.
- [I-D.ietf-tsvwg-iana-ports]
Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S.
Cheshire, "Internet Assigned Numbers Authority (IANA)
Procedures for the Management of the Service Name and
Transport Protocol Port Number Registry",
draft-ietf-tsvwg-iana-ports-10 (work in progress),
February 2011.
- [I-D.mdt-software-map-deployment]
Sun, Q., Chen, M., Chen, G., Sun, C., Tsou, T., and S.
Perreault, "Mapping of Address and Port (MAP) - Deployment
Considerations", draft-mdt-software-map-deployment-02
(work in progress), June 2012.
- [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.murakami-software-4rd]
Murakami, T., Troan, O., and S. Matsushima, "IPv4 Residual

Deployment on IPv6 infrastructure - protocol specification", draft-murakami-softwire-4rd-01 (work in progress), September 2011.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

Authors' Addresses

Tomasz Mrugalski
Internet Systems Consortium, Inc.
950 Charter Street
Redwood City, CA 94063
USA

Phone: +1 650 423 1345
Email: tomasz.mrugalski@gmail.com
URI: <http://www.isc.org/>

Ole Troan
Cisco Systems, Inc.
Telemarksvingen 20
Oslo N-0655
Norway

Email: ot@cisco.com
URI: <http://cisco.com>

Congxiao Bao
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN

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

Wojciech Dec
Cisco Systems, Inc.
The Netherlands

Phone:

Fax:

Email: wdec@cisco.com

URI:

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 2, 2012

O. Troan
cisco
S. Matsushima
SoftBank Telecom
T. Murakami
IP Infusion
X. Li
C. Bao
CERNET Center/Tsinghua
University
January 30, 2012

Mapping of Address and Port (MAP)
draft-mdt-softwire-mapping-address-and-port-03

Abstract

This document describes a generic mechanism for mapping between IPv4 addresses and IPv6 addresses and transport layer ports.

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 August 2, 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. Introduction	3
2. Conventions	4
3. Terminology	5
4. Architecture	6
5. Mapping Rules	7
5.1. Port mapping algorithm	8
5.1.1. Bit Representation of the Algorithm	9
5.1.2. GMA examples	9
5.1.3. GMA Provisioning Considerations	10
5.2. Basic mapping rule (BMR)	10
5.3. Forwarding mapping rule (FMR)	13
5.4. Default mapping rule (DMR)	14
6. The IPv6 Interface Identifier	15
7. IANA Considerations	16
8. Security Considerations	16
9. Contributors	16
10. Acknowledgements	16
11. References	17
11.1. Normative References	17
11.2. Informative References	17
Authors' Addresses	19

1. Introduction

The mechanism of mapping IPv4 addresses in IPv6 addresses has been described in numerous mechanisms dating back to 1996 [RFC1933]. The Automatic tunneling mechanism described in RFC1933, assigned a globally unique IPv6 address to a host by combining the host's IPv4 address with a well-known IPv6 prefix. Given an IPv6 packet with a destination address with an embedded IPv4 address, a node could automatically tunnel this packet by extracting the IPv4 tunnel end-point address from the IPv6 destination address.

There are numerous variations of this idea, described in 6over4 [RFC2529], 6to4 [RFC3056], ISATAP [RFC5214], and 6rd [RFC5969]. The differences between these are the use of well-known IPv6 prefixes, or Service Provider assigned IPv6 prefixes, and the position of the embedded IPv4 bits in the IPv6 address. Teredo [RFC4380] added a twist to this to achieve NAT traversal by also encoding transport layer ports into the IPv6 address. 6rd, to achieve more efficient encoding, allowed for only the suffix of an IPv4 address to be embedded, with the IPv4 prefix being deduced from other provisioning mechanisms.

NAT-PT [RFC2766](deprecated) combined with a DNS ALG used address mapping to put NAT state, namely the IPv6 to IPv4 binding encoded in an IPv6 address. This characteristic has been inherited by NAT64 [RFC6146] and DNS64 [RFC6147] which rely on an address format defined in [RFC6052]. [RFC6052] specifies the algorithmic translation of an IPv6 address to IPv4 address. In particular, [RFC6052] specifies the address format to build IPv4-converted and IPv4-translatable IPv6 addresses. RFC6052 discusses the transport of the port-set information in an IPv4-embedded IPv6 address but the conclusion was the following (excerpt from [RFC6052]):

"There have been proposals to complement stateless translation with a port range feature. Instead of mapping an IPv4 address to exactly one IPv6 prefix, the options would allow several IPv6 nodes to share an IPv4 address, with each node managing a different set of ports. If a port-set extension is needed, it could be defined later, using bits currently reserved as null in the suffix."

The commonalities of all these IPv6 over IPv4 mechanisms are:

- o Automatically provisions an IPv6 address for a host or an IPv6 prefix for a site
- o Algorithmic or implicit address resolution for tunneling or encapsulation. Given an IPv6 destination address, an IPv4 tunnel endpoint address can be calculated. Likewise for translation, an

IPv4 address can be calculated from an IPv6 destination address and vice versa.

- o Embedding of an IPv4 address or part thereof and optionally transport layer ports into an IPv6 address.

In phases of IPv4 to IPv6 migration, IPv6 only networks will be common, while there will still be a need for residual IPv4 deployment. This document describes a more generic mapping of IPv4 to IPv6 that can be used both for encapsulation (IPv4 over IPv6) and for translation between the two protocols.

Just as the IPv6 over IPv4 mechanisms referred to above, the residual IPv4 over IPv6 mechanisms must be capable of:

- o Provisioning an IPv4 prefix, an IPv4 address or a shared IPv4 address.
- o Algorithmically map between an IPv4 prefix, IPv4 address or a shared IPv4 address and an IPv6 address.

The unified mapping scheme described here supports translation mode, encapsulation mode, in both mesh and hub and spoke topologies.

This document describes delivery of IPv4 unicast service across an IPv6 infrastructure. IPv4 multicast is not considered further in this document.

The A+P (Address and Port) architecture of sharing an IPv4 address by distributing the port space is described in [RFC6346]. Specifically section 4 of [RFC6346] covers stateless mapping. The corresponding stateful solution DS-lite is described in [RFC6333]. The motivation for work is described in [I-D.ietf-software-stateless-4v6-motivation].

A companion document defines a DHCPv6 option for provisioning of MAP [I-D.mdt-software-map-dhcp-option]. Deployment considerations are described in [I-D.mdt-software-map-deployment].

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 RFC 2119 [RFC2119].

3. Terminology

MAP domain:	A set of MAP CEs and BRs connected to the same virtual link. A service provider may deploy a single MAP domain, or may utilize multiple MAP domains.
MAP Rule	A set of parameters describing the mapping between an IPv4 prefix, IPv4 address or shared IPv4 address and an IPv6 prefix or address. Each MAP node in the domain has the same set of rules.
MAP node	A device that implements MAP.
MAP Border Relay (BR):	A MAP enabled router managed by the service provider at the edge of a MAP domain. A Border Relay router has at least an IPv6-enabled interface and an IPv4 interface connected to the native IPv4 network. A MAP BR may also be referred to simply as a "BR" within the context of MAP.
MAP Customer Edge (CE):	A device functioning as a Customer Edge router in a MAP deployment. A typical MAP CE adopting MAP rules will serve a residential site with one WAN side interface, and one or more LAN side interfaces. A MAP CE may also be referred to simply as a "CE" within the context of MAP.
Port-set:	Each node has a separate part of the transport layer port space; denoted as a port-set.
Port-set ID (PSID):	Algorithmically identifies a set of ports exclusively assigned to the CE.
Shared IPv4 address:	An IPv4 address that is shared among multiple CEs. Only ports that belong to the assigned port-set can be used for communication. Also known as a Port-Restricted IPv4 address.
End-user IPv6 prefix:	The IPv6 prefix assigned to an End-user CE by other means than MAP itself. E.g. provisioned using DHCPv6 PD [RFC3633] or configured manually. It is unique for each CE.

MAP IPv6 address:	The IPv6 address used to reach the MAP function of a CE from other CE's and from BR's.
Rule IPv6 prefix:	An IPv6 prefix assigned by a Service Provider for a mapping rule.
Rule IPv4 prefix:	An IPv4 prefix assigned by a Service Provider for a mapping rule.
IPv4 Embedded Address (EA) bits:	The IPv4 EA-bits in the IPv6 address identify an IPv4 prefix/address (or part thereof) or a shared IPv4 address (or part thereof) and a port-set identifier.
MRT:	MAP Rule table. Address and Port aware datastructure, supporting longest match lookups. The MRT is used by the MAP forwarding function.

4. Architecture

A full IPv4 address or IPv4 prefix can be used like today, e.g. for identifying an interface or as a DHCP pool. A shared IPv4 address on the other hand, MUST NOT be used to identify an interface. While it is theoretically possible to make host stacks and applications port-aware, that is considered a too drastic change to the IP model [RFC6250].

The MAP architecture described here, restricts the use of the shared IPv4 address to only be used as the global address (outside) of the NAPT [RFC2663] running on the CE. The NAPT MUST in turn be connected to a MAP aware forwarding function, that does encapsulation/decapsulation or translation to IPv6.

For packets outbound from the private IPv4 network, the CE NAPT MUST translate transport identifiers (e.g. TCP and UDP port numbers) so that they fall within the assigned CE's port-range.

The forwarding function uses the MRT to make forwarding decisions. The table consist of the mapping rules. An entry in the table consists of an IPv4 prefix and PSID. The normal best matching prefix algorithm is used. With a maximum key length of 48 (32 + 16). E.g. with a sharing ratio of 64 (6 bit PSID length) a host route for this CE would be a /38 (32 + 6).

5. Mapping Rules

A MAP node is provisioned with one or more mapping rules.

Mapping rules are used differently depending on their function. Every MAP node must be provisioned with a Basic mapping rule. This is used by the node to configure itself with an IPv4 address, IPv4 prefix or shared IPv4 address from an End-user IPv6 prefix. This same basic rule can also be used for forwarding, where an IPv4 destination address and optionally a destination port is mapped into an IPv6 address or prefix. Additional mapping rules can be specified to allow for e.g. multiple different IPv4 subnets to exist within the domain. Additional mapping rules are recognized by having a Rule IPv6 prefix different from the base End-user IPv6 prefix.

Traffic outside of the domain (IPv4 address not matching (using longest matching prefix) any Rule IPv4 prefix in the Rules database) will be forward using the Default mapping rule. The Default mapping rule maps outside destinations to the BR's IPv6 address or prefix.

There are three types of mapping rules:

1. Basic Mapping Rule - used for IPv4 prefix, address or port set assignment. There can only be one Basic Mapping Rule per End-user IPv6 prefix. The Basic Mapping Rule is used to configure the MAP IPv6 address or prefix.
 - * Rule IPv6 prefix (including prefix length)
 - * Rule IPv4 prefix (including prefix length)
 - * Rule EA-bits length (in bits)
 - * Rule Port Parameters (optional)
2. Forwarding Mapping Rule - used for forwarding. The Basic Mapping Rule is also a Forwarding Mapping Rule. Each Forwarding Mapping Rule will result in an entry in the MRT for the Rule IPv4 prefix.
 - * Rule IPv6 prefix (including prefix length)
 - * Rule IPv4 prefix (including prefix length)
 - * Rule EA-bits length (in bits)
 - * Rule Port Parameters (optional)

3. Default Mapping Rule - used for destinations outside the MAP domain. A 0.0.0.0/0 entry is installed in the MRT for this rule.

- * Rule IPv6 prefix (including prefix length)

- * Rule BR IPv4 address

A MAP node finds its Basic Mapping Rule by doing a longest match between the End-user IPv6 prefix and the Rule IPv6 prefix in the Mapping Rule database. The rule is then used for IPv4 prefix, address or shared address assignment.

A MAP IPv6 address (or prefix) is formed from the BMR Rule IPv6 prefix. This address MUST be assigned to an interface of the MAP node and is used to terminate all MAP traffic being sent or received to the node.

Port-aware IPv4 entries in the MRT are installed for all the Forwarding Mapping Rules and an IPv4 default route for the Default Mapping Rule.

In hub and spoke mode, all traffic MUST be forwarded using the Default Mapping Rule.

5.1. Port mapping algorithm

Different Port-Set Identifiers (PSID) MUST have non-overlapping port-sets. The two extreme cases are: (1) the port numbers are not contiguous for each PSID, but uniformly distributed across the port range (0-65535); (2) the port numbers are contiguous in a single range for each PSID. The port mapping algorithm proposed here is called the Generalized Modulus Algorithm (GMA) and supports both these cases.

For a given sharing ratio (R) and the maximum number of contiguous ports (M), the GMA algorithm is defined as:

1. The port number (P) of a given PSID (K) is composed of:

$$P = R * M * j + M * K + i$$

Where:

- * PSID: $K = 0$ to $R - 1$

- * Port range index: $j = (4096 / M) / R$ to $((65536 / M) / R) - 1$, if the port numbers (0 - 4095) are excluded.

* Contiguous Port index: $i = 0$ to $M - 1$

2. The PSID (K) of a given port number (P) is determined by:

$$K = (\text{floor}(P/M)) \% R$$

Where:

* $\%$ is the modulus operator

* $\text{floor}(\text{arg})$ is a function that returns the largest integer not greater than arg .

5.1.1. Bit Representation of the Algorithm

Given a sharing ratio ($R=2^k$), the maximum number of contiguous ports ($M=2^m$), for any PSID (K) and available ports (P) can be represented as:

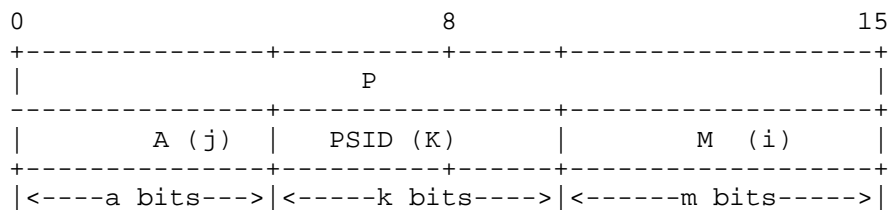


Figure 1: Bit representation

Where j and i are the same indexes defined in the port mapping algorithm.

For any port number, the PSID can be obtained by bit mask operation.

For $a > 0$, j MUST be larger than 0. This ensures that the algorithm excludes the system ports ([I-D.ietf-tsvwg-iana-ports]). For $a = 0$, j MAY be 0 to allow for the provisioning of the system ports.

5.1.2. GMA examples

For example, for $R = 1024$, PSID offset: $a = 4$ and PSID length: $k = 10$ bits

	Port-set-1	Port-set-2
PSID=0	4096, 4097, 4098, 4099,	8192, 8193, 8194, 8195, ...
PSID=1	4100, 4101, 4102, 4103,	8196, 8197, 8198, 8199, ...
PSID=2	4104, 4105, 4106, 4107,	8200, 8201, 8202, 8203, ...
PSID=3	4108, 4109, 4110, 4111,	8204, 8205, 8206, 8207, ...
...		
PSID=1023	8188, 8189, 8190, 8191,	12284, 12285, 12286, 12287, ...

Example 1: with offset = 4 ($a = 4$)

For example, for $R = 64$, $a = 0$ (PSID offset = 0 and PSID length = 6 bits):

	Port-set
PSID=0	[0 - 1023]
PSID=1	[1024 - 2047]
PSID=2	[2048 - 3071]
PSID=3	[3072 - 4095]
...	
PSID=63	[64512 - 65535]

Example 2: with offset = 0 ($a = 0$)

5.1.3. GMA Provisioning Considerations

The number of offset bits (a) and excluded ports are optionally provisioned via the "Rule Port Mapping Parameters" in the Basic Mapping Rule.

The defaults are:

- o Excluded ports : 0-4095
- o Offset bits (a) : 4

To simplify the GMA port mapping algorithm the defaults are chosen so that the PSID field starts on a nibble boundary and the excluded port range (0-1023) is extended to 0-4095.

5.2. Basic mapping rule (BMR)

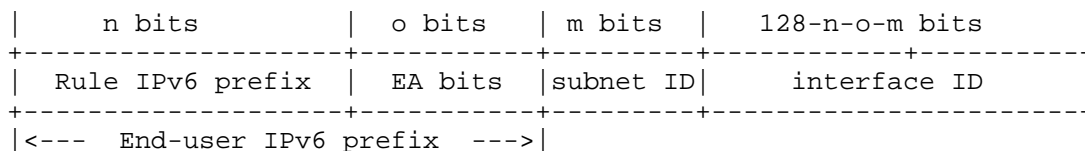


Figure 2: IPv6 address format

The Embedded Address bits (EA bits) are unique per end user within a Rule IPv6 prefix. The Rule IPv6 prefix is the part of the End-user IPv6 prefix that is common among all CEs using the same Basic Mapping Rule within the MAP domain. The EA bits encode the CE specific IPv4 address and port information. The EA bits can contain a full or part of an IPv4 prefix or address, and in the shared IPv4 address case contains a Port-Set Identifier (PSID).

The MAP IPv6 address is created by concatenating the End-user IPv6 prefix with the MAP subnet-id and the interface-id as specified in Section 6.

The MAP subnet ID is defined to be the first subnet (all bits set to zero). A MAP node MUST reserve the first IPv6 prefix in a End-user IPv6 prefix for the purpose of MAP.

Shared IPv4 address:

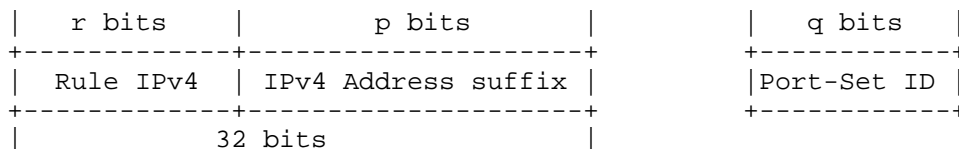


Figure 3: Shared IPv4 address

Complete IPv4 address:

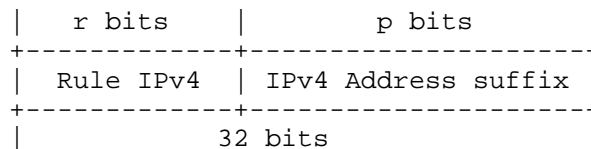


Figure 4: Complete IPv4 address

IPv4 prefix:

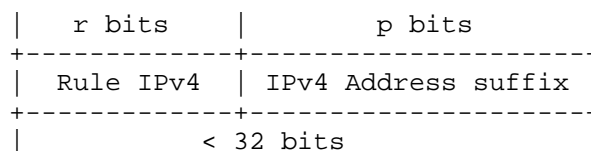


Figure 5: IPv4 prefix

The length of r MAY be zero, in which case the complete IPv4 address or prefix is encoded in the EA bits. If only a part of the IPv4 address/prefix is encoded in the EA bits, the Rule IPv4 prefix is provisioned to the CE by other means (e.g. a DHCPv6 option). To create a complete IPv4 address (or prefix), the IPv4 address suffix (p) from the EA bits, are concatenated with the Rule IPv4 prefix (r bits).

The offset of the EA bits field in the IPv6 address is equal to the BMR Rule IPv6 prefix length. The length of the EA bits field (o) is given by the BMR Rule EA-bits length. The sum of the Rule IPv6 Prefix length and the Rule EA-bits length MUST be less or equal than the End-user IPv6 prefix length.

If $o + r < 32$ (length of the IPv4 address in bits), then an IPv4 prefix is assigned.

If $o + r$ is equal to 32, then a full IPv4 address is to be assigned. The address is created by concatenating the Rule IPv4 prefix and the EA-bits.

If $o + r$ is > 32 , then a shared IPv4 address is to be assigned. The number of IPv4 address suffix bits (p) in the EA bits is given by $32 - r$ bits. The PSID bits are used to create a port-set. The length of the PSID bit field within EA bits is: $o - p$.

In the following examples, only the suffix (last 8 bits) of the IPv4 address is embedded in the EA bits ($r = 24$), while the IPv4 prefix (first 24 bits) is given in the BMR Rule IPv4 prefix.

Example:

Given:

```
End-user IPv6 prefix: 2001:db8:0012:3400::/56
Basic Mapping Rule:  {2001:db8:0000::/40 (Rule IPv6 prefix),
                      192.0.2.0/24 (Rule IPv4 prefix),
                      16 (Rule EA-bits length)}
Sharing ratio:       256 (16 - (32 - 24) = 8. 2^8 = 256)
PSID offset:         4
```

We get IPv4 address and port-set:

```
EA bits offset:      40
IPv4 suffix bits (p): Length of IPv4 address (32) -
                      IPv4 prefix length (24) = 8
IPv4 address:        192.0.2.18

PSID start:          40 + p = 40 + 8 = 48
PSID length:         o - p = 16 (56 - 40) - 8 = 8
PSID:                0x34
Port-set-1:          4928, 4929, 4930, 4931, 4932, 4933, 4934, 4935,
                      4936, 4937, 4938, 4939, 4940, 4941, 4942, 4943
Port-set-2:          9024, 9025, 9026, 9027, 9028, 9029, 9030, 9031,
                      9032, 9033, 9034, 9035, 9036, 9037, 9038, 9039
...
Port-set-15:         62272, 62273, 62274, 62275,
                      62276, 62277, 62278, 62279,
                      62280, 62281, 62282, 62283,
                      62284, 62285, 62286, 62287,
```

5.3. Forwarding mapping rule (FMR)

On adding an FMR rule, an IPv4 route is installed in the AP RIB for the Rule IPv4 prefix.

On forwarding an IPv4 packet, a best matching prefix lookup is done in the IPv4 routing table and the correct FMR is chosen.

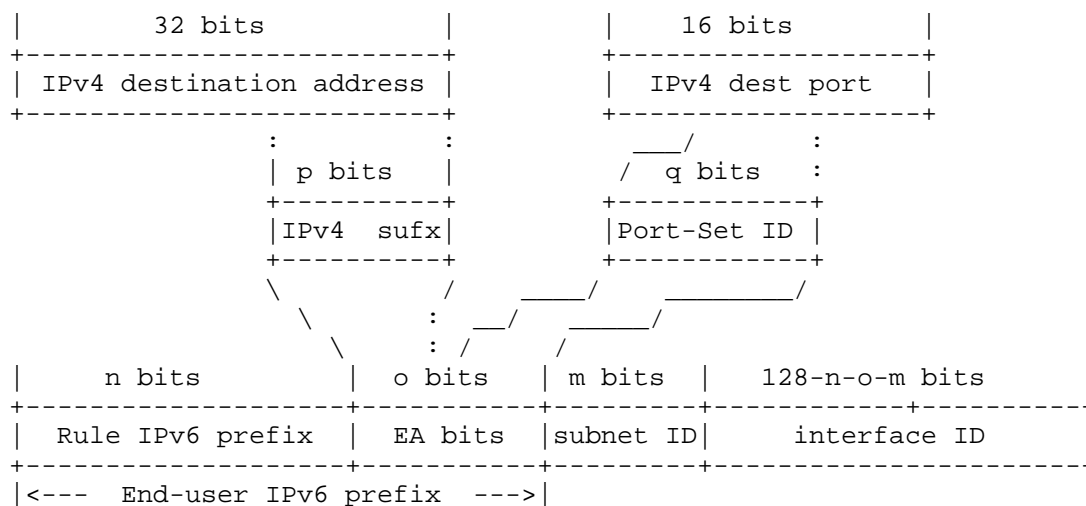


Figure 6: Deriving of MAP IPv6 address

Example:

Given:

IPv4 destination address: 192.0.2.18

IPv4 destination port: 9030

Forwarding Mapping Rule: {2001:db8:0000::/40 (Rule IPv6 prefix),
192.0.2.0/24 (Rule IPv4 prefix),
16 (Rule EA-bits length)}

PSID offset: 4

We get IPv6 address:

IPv4 suffix bits (p): 32 - 24 = 8 (18 (0x12))

PSID length: 8

PSID: 0x34 (9030 (0x2346))

EA bits: 0x1234

MAP IPv6 address: 2001:db8:0012:3400:00c0:0002:1200:3400

5.4. Default mapping rule (DMR)

The Default Mapping rule is used to reach IPv4 destinations outside of the MAP domain. Traffic using this rule will be sent from a CE to a BR.

The Rule IPv4 prefix in the DMR is: 0.0.0.0/0. The Rule IPv6 prefix is the IPv6 address or prefix of the BR. Which is used, is dependent on the mode used. For example translation requires that the IPv4 destination address is encoded in the BR IPv6 address, so only a

prefix is used in the DMR to allow for a generated interface identifier. For the encapsulation mode the Rule IPv6 prefix can be the full IPv6 address of the BR.

There MUST be only one Default Mapping Rule within a MAP domain.

Default Mapping Rule:

```
{2001:db8:0001:0000:<interface-id>:/128 (Rule IPv6 prefix),  
 0.0.0.0/0 (Rule IPv4 prefix),  
 192.0.2.1 (BR IPv4 address)}
```

Example 3: Default Mapping Rule

In most implementations of a routing table, the next-hop address must be of the same address family as the prefix. To satisfy this requirement a BR IPv4 address is included in the rule. Giving a default route in the IPv4 routing table:

```
0.0.0.0 -> 192.0.2.1, MAP-Interface0
```

6. The IPv6 Interface Identifier

The Interface identifier format is based on the format specified in section 2.2 of [RFC6052], with the added PSID format field.

In an encapsulation solution, an IPv4 address and port is mapped to an IPv6 address. This is the address of the tunnel end point of the receiving MAP CE. For traffic outside the MAP domain, the IPv6 tunnel end point address is the IPv6 address of the BR. The interface-id used for all MAP nodes in the domain MUST be deterministic.

When translating, the destination IPv4 address is translated into a corresponding IPv6 address. In the case of traffic outside of the MAP domain, it is translated to the BR's IPv6 prefix. For the BR to be able to reverse the translation, the full destination IPv4 address must be encoded in the IPv6 address. The same thing applies if an IPv4 prefix is encoded in the IPv6 address, then the reverse translator needs to know the full destination IPv4 address, which has to be encoded in the interface-id.

The encoding of the full IPv4 address into the interface identifier, both for the source and destination IPv6 addresses have been shown to be useful for troubleshooting.

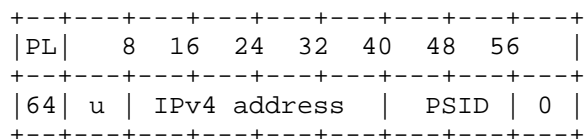


Figure 7

In the case of an IPv4 prefix, the IPv4 address field is right-padded with zeroes up to 32 bits. The PSID field is left-padded to create a 16 bit field. For an IPv4 prefix or a complete IPv4 address, the PSID field is zero.

If the End-user IPv6 prefix length is larger than 64, the most significant parts of the interface identifier is overwritten by the prefix. For translation mode the End-user IPv6 prefix MUST be 64 or shorter.

7. IANA Considerations

This specification does not require any IANA actions.

8. Security Considerations

Specific security considerations with the MAP mechanism are detailed in the encapsulation and translation documents [I-D.mdt-map-t/I-D.mdt-map-e].

[RFC6269] outlines general issues with IPv4 address sharing.

9. Contributors

Mohamed Boucadair, Gang Chen, Maoke Chen, Wojciech Dec, Xiaohong Deng, Jouni Korhonen, Tomasz Mrugalski, Jacni Qin, Chunfa Sun, Qiong Sun, Leaf Yeh.

10. Acknowledgements

This document is based on the ideas of many. In particular Remi Despres, who has tirelessly worked on generalized mechanisms for stateless address mapping.

The authors would like to thank Guillaume Gottard, Dan Wing, Jan

Zorz, Necj Scoberne, Tina Tsou for their thorough review and comments.

11. References

11.1. Normative References

- [I-D.mdt-software-map-dhcp-option]
Mrugalski, T., Boucadair, M., Deng, X., Troan, O., and C. Bao, "DHCPv6 Options for Mapping of Address and Port", draft-mdt-software-map-dhcp-option-02 (work in progress), January 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.

11.2. Informative References

- [I-D.ietf-software-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-ietf-software-stateless-4v6-motivation-00 (work in progress), September 2011.
- [I-D.ietf-tsvwg-iana-ports]
Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", draft-ietf-tsvwg-iana-ports-10 (work in progress), February 2011.
- [RFC1933] Gilligan, R. and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 1933, April 1996.
- [RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, March 1999.
- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.
- [RFC2766] Tsirtsis, G. and P. Srisuresh, "Network Address

- Translation - Protocol Translation (NAT-PT)", RFC 2766, February 2000.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.
- [RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.
- [RFC6250] Thaler, D., "Evolution of the IP Model", RFC 6250, May 2011.
- [RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

Authors' Addresses

Ole Troan
cisco
Oslo
Norway

Email: ot@cisco.com

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

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

Tetsuya Murakami
IP Infusion
1188 East Arques Avenue
Sunnyvale
USA

Email: tetsuya@ipinfusion.com

Xing Li
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN

Email: xing@cernet.edu.cn

Congxiao Bao
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN

Email: congxiao@cernet.edu.cn

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 12, 2012

R. Penno
A. Durand
Juniper Networks
A. Clauberg
Deutsche Telekom AG
L. Hoffmann
Bouygues Telecom
March 11, 2012

Stateless DS-Lite
draft-penno-softwire-sdnat-02

Abstract

This memo define a simple stateless and deterministic mode of operating a carrier-grade NAT as a backward compatible evolution of DS-Lite.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 12, 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. Introduction	3
2. Stateless DS-Lite CPE	4
2.1. Learning external IPv4 address	4
2.2. Learning external port range	4
2.3. Stateless DS-Lite CPE operation	5
2.4. Host-based Stateless DS-Lite	5
3. Stateless AFTR	5
3.1. Anycast IPv6 address for Stateless AFTR	5
3.2. Stateless AFTR IPv4 address pool	5
3.3. Stateless AFTR per-subscriber mapping table	5
3.4. Stateless AFTR decapsulation rules	6
3.5. Stateless AFTR encapsulation rules	6
3.6. Redundancy and fail over	7
3.7. SD-AFTR stateless domain	7
4. Backward compatibility with DS-Lite	7
5. ICMP port restricted message	8
5.1. Introduction	8
5.2. Source port restricted ICMP	8
5.3. Host behavior	9
6. IANA Considerations	9
7. Security Considerations	9
8. References	10
8.1. Normative references	10
8.2. Informative references	10
Authors' Addresses	11

1. Introduction

DS-Lite [RFC6333], is a solution to deal with the IPv4 exhaustion problem once an IPv6 access network is deployed. It enables unmodified IPv4 application to access the IPv4 Internet over the IPv6 access network. In the DS-Lite architecture, global IPv4 addresses are shared among subscribers in the AFTR, acting as a Carrier-Grade NAT (CGN).

[I-D.ietf-softwire-public-4over6] extends the original DS-Lite model to offer a mode where the NAT function is performed in the CPE. This simplifies the AFTR operation as it does not have to perform the NAT function anymore, however, the flip side is that the address sharing function among subscribers was no longer available.

[I-D.cui-softwire-b4-translated-ds-lite] introduces port restrictions, but does not completely specifies how the CPE acquires the information about its IPv4 address and its port range. More importantly, that draft does not explain how this solution can be deployed in a regular DS-Lite environment. This memo addresses these issues and clarifies the operation model.

Other approaches like variations of 4rd allows also for a full stateless operation of the decapsulation device. By introducing a strong coupling between the IPv6 address and the derived IPv4 address, they get rid of the per-subscriber state on the decapsulation devices. The approach take here argues that such per-subscriber state is not an issue as it is easily replicated among all decapsulation devices. Eliminating the strong coupling between IPv6 and IPv4 derived addresses, the approach presented here enables service providers a greater flexibility on how their limited pool of IPv4 addresses is managed. It also provide greater freedom on how IPv6 addresses are allocated, as sequential allocation is no longer a pre-requisite.

The approach presented here is stateless and deterministic. It is stateless is NAT bindings are maintained on the CPE, not on the AFTR. It is deterministic as no logs are required on the AFTR to identify which subscriber is using an external Ipv4 address and port.

The stateless DS-Lite architecture has the following characteristics:

- o Backward compatible with DS-Lite. A mix of regular DS-Lite CPE and stateless DS-Lite CPEs can interoperate with a stateless DS-Lite AFTR.
- o Zero log: Because the AFTR relies only on a per-subscriber mapping table that is reversible, the ISP does not need to keep any NAT binding logs.

- o Stateless AFTR: There is no per-session state on the AFTRs. By leveraging this stateless and deterministic mode of operation, an ISP can deploy any number of AFTRs to provide redundancy and scalability at low cost. Because there is no per-flow state to maintain, AFTR can implement the functionality in hardware and perform it at high speed with low latency.
- o Flexibility of operation: The ISP can add or remove addresses from the NAT pool without having to renumber the access network.
- o Leverage IPv6: This stateless DS-Lite model leverage the IPv6 access network deployed by the ISPs.

2. Stateless DS-Lite CPE

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].

A Stateless DS-Lite CPE operates in similar fashion than a regular DS-Lite CPE, where the NAT function is re-introduced in CPE with a modification on how ports are managed.

2.1. Learning external IPv4 address

A stateless DS-Lite CPE MUST implement the DHCPv4 client relay option defined in [I-D.ietf-dhc-dhcpv4-over-ipv6] to learn its external IPv4 address. Other mechanism, such as manual configuration or TR69, MAY be implemented.

2.2. Learning external port range

A stateless DS-Lite CPE MUST implement the ICMP "port restricted" option defined later in this memo.

At boot time and later at intervals of 1h +/- a random number of seconds between 0 and 900), the stateless DS-Lite CPE MUST send packets with source port 0, source IPv4 address of the B4 element, destination IPv4 address 192.0.0.1 (the AFTR well-known IPv4 address) destination port 0, for each of the supported transport protocols (usually TCP and UDP). This will trigger an ICMP "port restricted" message from the AFTR.

After validating the content of the "ICMP port restricted" message, the stateless DS-Lite CPE MUST configure its port pool with it. If existing connections were using source ports outside of that range, the stateless DS-Lite CPE MUST terminate them.

2.3. Stateless DS-Lite CPE operation

The stateless DS-Lite CPE performs IPv4 NAT from the internal RFC1918 addresses to the IPv4 address configured on the WAN interface, restricting its available ports to the range obtained as described above.

2.4. Host-based Stateless DS-Lite

Any host initiating directly a DS-Lite IPv4 over IPv6 tunnel can benefit from this techniques by implementing a 'virtual' stateless DS-Lite CPE function within its IP stack.

3. Stateless AFTR

3.1. Anycast IPv6 address for Stateless AFTR

All stateless AFTRs associated to a domain (or group of subscribers) will be configured with the same IPv6 address on the interface facing IPv6 subscribers. A route for that IPv6 address will be anycasted within the access network.

3.2. Stateless AFTR IPv4 address pool

All stateless AFTRs associated to a domain (or group of subscribers) MUST be configured with the same pool of global IPv4 addresses.

Routes to the pool of global IPv4 addresses configured on the stateless AFTRs will be anycasted by the relevant AFTRs within the ISP routing domain.

3.3. Stateless AFTR per-subscriber mapping table

Stateless AFTRs associated to a domain (or group of subscribers) MUST be configured with the same per-subscriber mapping table, associating the IPv6 address of the subscriber CPE to the external IPv4 address and port range provisioned for this subscriber.

Because the association IPv6 address --- IPv4 address + port range is not tied to a mathematical formula, the ISP maintains all flexibility to allocate independently IPv6 address and IPv4 addresses. In particular, IPv6 addresses do not have to be allocated sequentially and IPv4 resources can be modified freely.

IPv6 address	IPv4 address	port-range
2001:db8::1	1.2.3.4	1000-1999
2001:db8::5	1.2.3.4	2000-2999
2001:db8::a:1	1.2.3.4	3000-3999

Figure 1: Per-subscriber mapping table example

This per-subscriber mapping table can be implemented in various ways which details are out of scope for this memo. In its simplest form, it can be a static file that is replicated out-of-band on the AFTRs. In a more elaborated way, this table can be dynamically built using radius queries to a subscriber database.

3.4. Stateless AFTR decapsulation rules

Upstream IPv4 over IPv6 traffic will be decapsulated by the AFTR. The AFTR MUST check the outer IPv6 source address belongs to an identified subscriber and drop the traffic if not. The AFTR MUST then check the inner IPv4 header to make sure the IPv4 source address and ports are valid according to the per-subscriber mapping table.

If the inner IPv4 source address does not match the entry in the per-subscriber mapping table, the packet MUST be discarded and an ICMP 'administratively prohibited' message MAY be returned.

If the IPv4 source port number falls outside of the range allocated to the subscriber, the AFTR MUST discard the datagram and MUST send back an ICMP "port restricted" message to the IPv6 source address of the packet.

Fragmentation and reassembly is treated as in DS-Lite [RFC6333].

3.5. Stateless AFTR encapsulation rules

Downstream traffic is validated using the per-subscriber mapping table. Traffic that falls outside of the IPv4 address/port range entries in that table MUST be discarded. Validated traffic is then encapsulated in IPv6 and forwarded to the associated IPv6 address.

Fragmentation and reassembly is treated as in DS-Lite [RFC6333].

3.6. Redundancy and fail over

Because there is no per-flow state, upstream and downstream traffic can use any stateless AFTR.

3.7. SD-AFTR stateless domain

Using the DHCPv6 DS-Lite tunnel-end-point option, groups of subscribers can be associated to a different stateless AFTR domain. That can allow for differentiated level of services, e.g. number of ports per customer device, QoS, bandwidth, value added services,...

4. Backward compatibility with DS-Lite

A number of service providers are, or are in the process of, deploying DS-Lite in their network. They are interested in evolving their design toward a stateless model. Backward compatibility is a critical issue, as, from an operational perspective, it is difficult to get all CPEs evolve at the same time.

So AFTRs have to be ready to service CPEs that are pure DS-Lite, some that are implementing only DHCPv4 over IPv6 and handle the NAT on the full IPv4 address themselves and some that also implement port restrictions via the ICMP message described here. For this reason, a AFTR operating in backward compatibility mode MAY decide to re-NAT upstream packets which source port number do not fall into the predefined range instead of simply dropping the packets.

The operating model is the following:

- o Stateless DS-Lite: for CPEs that pre-NAT and pre-shape the source port space into the range assigned to the subscriber: decapsulate, check per-subscriber mapping, forward.
- o B4-translated DS-Lite: for CPEs that performs NAT before encapsulation and are allocated a full IPv4 address: decapsulate, check per-subscriber mapping, forward.
- o Re-shaper DS-Lite: for CPEs that pre NAT but fail to restrict the source ports: decapsulate, check per-subscriber mapping, re-NAT statefully the packets into the restricted port range, mark range as 'stateful', forward.
- o Regular DS-Lite: for regular DS-Lite CPEs that do not pre-NAT: decapsulate, NAT statefully, forward.

In such a backward compatibility mode, the AFTR is only operating statelessly for the stateless DS-Lite CPEs. It needs to maintain per-flow state for the regular DS-Lite CPEs and the non-ICMP port restricted compliant CPEs. In this legacy mode where per-flow state is required, the simple anycast-based fail-over mechanism is no longer available.

5. ICMP port restricted message

Note: this section may end-up being a separate Internet draft.

5.1. Introduction

In the framework of A+P RFC 6346 [RFC6346], sources may be restricted to use only a subset of the port range of a transport protocol associated with an IPv4 address. When that source transmit a packet with a source outside of the pre-authorized range, the upstream NAT will drop the packet and use the ICMP message defined here to inform the source of the actual port range allocated.

This memo defines such ICMP messages for TCP and UDP and leaves the definition of the ICMP option for other transport protocol for future work.

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].

5.2. Source port restricted ICMP

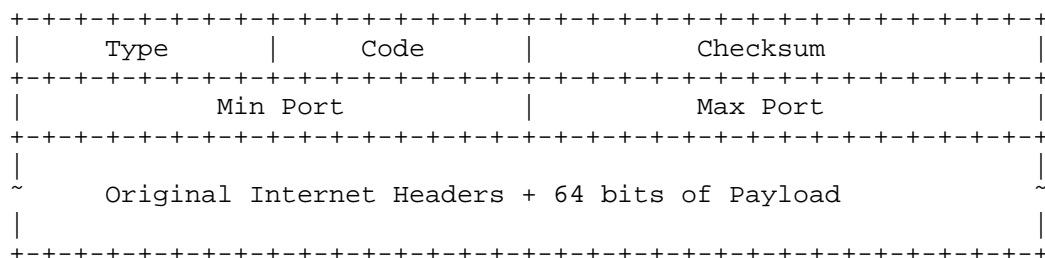


Figure 2: Source Port Restricted ICMP

Type: TBD for Source Port Restricted

Checksum: The checksum is the 16-bit ones's complement of the one's complement sum of the ICMP message starting with the ICMP Type. For

computing the checksum , the checksum field should be zero. This checksum may be replaced in the future.

Code: 6 for TCP, 17 for UDP

Min Port: The lowest port number allocated for that source.

Max Port: The highest port number allocated for that source.

5.3. Host behavior

A host receiving an ICMP type TBD message for a given transport protocol SHOULD NOT send packets sourced by the IP address(es) corresponding to the interface that received that ICMP message with source ports outside of the range specified for the given transport protocol.

Packets sourced with port numbers outside of the restricted range MAY be dropped or NATed upstream to fit within the restricted range.

A host MUST NOT take port restriction information applying to a given IP address and transport protocol and applies it to other IP addresses on other interfaces and/or other transport protocols.

If Min Port = 0 and Max Port = 65535, it indicates that the entire port range for the given transport protocol is available. If such 'full range' messages are received for all transport protocols, the host can take this as an indication that its IP address is probably not shared with other devices.

In order to mitigate possible man in the middle attacks, a host MUST discard ICMP type TBD messages if the associated port range (Max Port - Min Port) is lower than 64.

6. IANA Considerations

IANA is to allocated a code point for this ICMP message type.

7. Security Considerations

This ICMP message type has the same security properties as other ICMP messages such as Redirect or Destination Unreachable. A man-in-the-middle attack can be mounted to create a DOS attack on the source. Ingress filtering on network boundary can mitigate such attacks. However, in case such filtering measures are not enough, the additional provision that a host MUST discard such ICMP message with

a port range smaller than 64 can mitigate even further such attacks.

As described in [RFC6269], with any fixed size address sharing techniques, port randomization is achieved with a smaller entropy.

Recommendations listed in [RFC6302] applies.

8. References

8.1. Normative references

- [I-D.ietf-dhc-dhcpv4-over-ipv6]
Lemon, T., Cui, Y., Wu, P., and J. Wu, "DHCPv4 over IPv6 Transport", draft-ietf-dhc-dhcpv4-over-ipv6-00 (work in progress), November 2011.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

8.2. Informative references

- [I-D.cui-software-b4-translated-ds-lite]
Boucadair, M., Sun, Q., Tsou, T., Lee, Y., and Y. Cui, "Lightweight 4over6: An Extension to DS-Lite Architecture", draft-cui-software-b4-translated-ds-lite-05 (work in progress), February 2012.
- [I-D.ietf-pcp-base]
Cheshire, S., Boucadair, M., Selkirk, P., Wing, D., and R. Penno, "Port Control Protocol (PCP)", draft-ietf-pcp-base-23 (work in progress), February 2012.
- [I-D.ietf-software-public-4over6]
Cui, Y., Wu, J., Wu, P., Metz, C., Vautrin, O., and Y. Lee, "Public IPv4 over Access IPv6 Network", draft-ietf-software-public-4over6-00 (work in progress), September 2011.
- [RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269,

June 2011.

[RFC6302] Durand, A., Gashinsky, I., Lee, D., and S. Sheppard,
"Logging Recommendations for Internet-Facing Servers",
BCP 162, RFC 6302, June 2011.

[RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the
IPv4 Address Shortage", RFC 6346, August 2011.

Authors' Addresses

Reinaldo Penno
Juniper Networks
1194 North Mathilda Avenue
Sunnyvale, CA 94089-1206
USA

Email: rpenno@juniper.net

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

Email: adurand@juniper.net

Alex Clauberg
Deutsche Telekom AG
GTN-FM4
Landgrabenweg 151
Bonn, CA 53227
Germany

Email: axel.clauberg@telekom.de

Lionel Hoffmann
Bouygues Telecom
TECHNOPOLE
13/15 Avenue du Marechal Juin
Meudon 92360
France

Email: lhoffman@bouyguestelecom.fr

Softwire WG
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2012

J. Qin
ZTE
M. Boucadair
France Telecom
T. Tsou
Huawei Technologies (USA)
October 31, 2011

DHCPv6 Options for IPv6 DS-Lite Multicast Prefix
draft-qin-softwire-multicast-prefix-option-01

Abstract

This document defines Dynamic Host Configuration Protocol version 6 (DHCPv6) Options for multicast transition solutions, aiming to convey the IPv6 prefixes to be used to build unicast and multicast IPv4-embedded IPv6 addresses.

These options can be in particular used in the context of DS-Lite, Stateless A+P and other IPv4-IPv6 interconnection techniques.

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 May 3, 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	3
1.1. Requirements Language	3
2. Terminology	3
3. PREFIX64 DHCPv6 Option	4
3.1. Option Format	4
3.2. M_PREFIX64 Sub-option	4
3.3. U_PREFIX64 Sub-option	5
4. Client Behaviour	6
5. Server Behaviour	6
6. Security Considerations	7
7. Acknowledgements	7
8. IANA Considerations	7
9. References	8
9.1. Normative References	8
9.2. Informative References	8
Authors' Addresses	9

1. Introduction

[I-D.ietf-softwire-dslite-multicast] and several other solutions (e.g., [I-D.ietf-softwire-mesh-multicast], [I-D.venaas-behave-mcast46], etc.) are proposed for the delivery of multicast services in the context of transition to IPv6. Even these solutions may have different applicable use cases, they all use specific IPv6 addresses to embed IPv4 addresses, for both the multicast group addresses [I-D.boucadair-behave-64-multicast-address-format], and the multicast source addresses [RFC6052].

This document defines DHCPv6 options [RFC3315] to convey the IPv6 prefixes (a.k.a., PREFIX64) to be used for constructing these IPv4-embedded IPv6 addresses.

These options can be in particular used in the context of DS-Lite [RFC6333], Stateless A+P [RFC6346] and other IPv4-IPv6 interconnection techniques.

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:

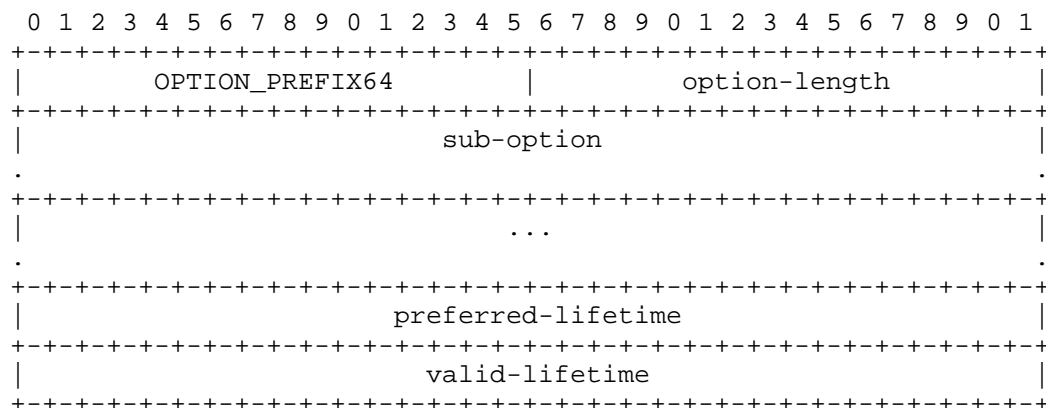
- o IPv4-embedded IPv6 address: is an IPv6 address which embeds a 32 bit-encoded IPv4 address [RFC6052]. An IPv4-embedded IPv6 address can be unicast or multicast address.
- o PREFIX64: is a dedicated IPv6 prefix for building IPv4-embedded IPv6 addresses. A PREFIX64 can be of unicast or multicast.
- o M_PREFIX64: denotes a multicast PREFIX64. It may belong to the SSM range (i.e., ff3x::/32 [RFC4607]) or ASM range.
- o U_PREFIX64: denotes a unicast PREFIX64 for building the IPv4-embedded IPv6 addresses of multicast sources in SSM mode.

3. PREFIX64 DHCPv6 Option

OPTION_PREFIX64 is defined to convey the IPv6 prefix(es) to use to synthesize IPv4-embedded IPv6 addresses. This option MAY enclose one or more sub-options.

3.1. Option Format

Figure 1 shows the format of the OPTION_PREFIX64 DHCPv6 option.



option-code: OPTION_PREFIX64 (TBD)

option-length: The length of enclosed sub-option(s) + 8 in octets

sub-option: One or several sub-options. Two sub-codes are defined in this document:

- (1) SUB_OPTION_M_PREFIX64
- (2) SUB_OPTION_U_PREFIX64

preferred-lifetime: The preferred lifetime for the IPv6 prefix(es) in the sub-option(s), expressed in units of seconds.

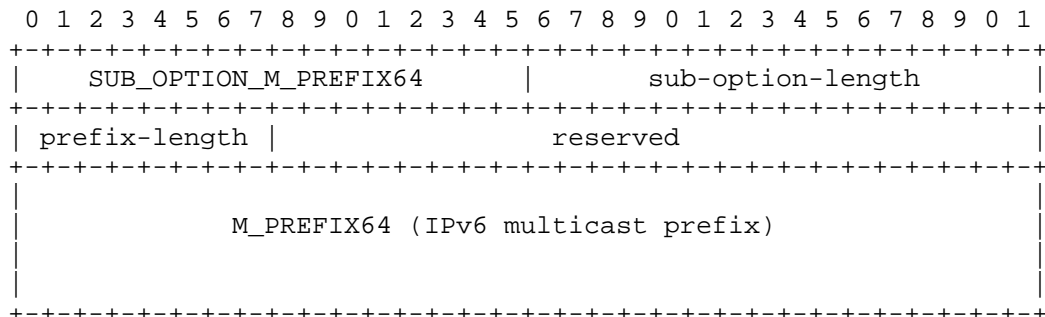
valid-lifetime: The valid lifetime for the IPv6 prefix(es) in the sub-option(s), expressed in units of seconds.

Figure 1: DHCPv6 Option Format for PREFIX64

3.2. M_PREFIX64 Sub-option

This sub-option (Figure 2) is defined to convey the IPv6 multicast prefix to use to synthesize the IPv4-embedded IPv6 addresses of the multicast groups [I-D.boucadair-behave-64-multicast-address-format]. The conveyed multicast IPv6 prefix MAY belong to the SSM range (i.e.,

ff3x::/32 [RFC4607]) or ASM range.



sub-option-code: SUB_OPTION_M_PREFIX64 (TBD)

sub-option-len: 20 in octets

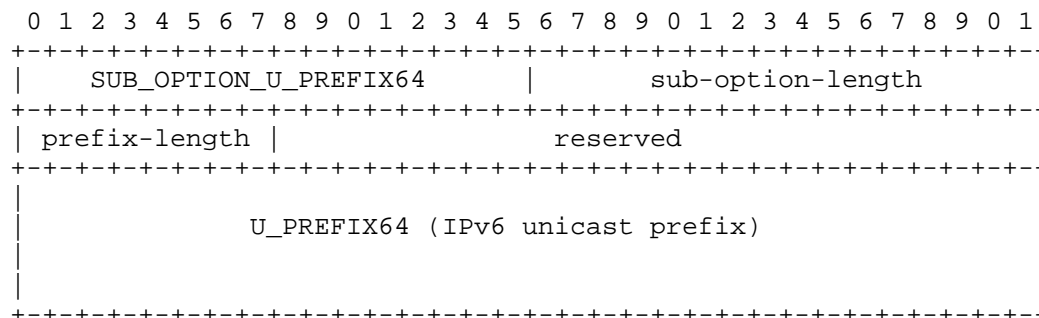
prefix-length: the length of M_PREFIX64 in bits

M_PREFIX64: the multicast prefix for constructing the IPv4-embedded IPv6 addresses of multicast groups. It MAY belong to SSM or ASM address range.

Figure 2: DHCPv6 Sub-option Format for M_PREFIX64

3.3. U_PREFIX64 Sub-option

This sub-option (Figure 3) is defined to convey the IPv6 unicast prefix to be used in SSM mode for constructing the IPv4-embedded IPv6 addresses of the multicast sources. It is also used to extract the IPv4 address from received multicast data flows (e.g., [I-D.ietf-softwire-dslite-multicast]). The address synthesis MUST follow the guidelines documented at [RFC6052].



sub-option-code: SUB_OPTION_U_PREFIX64 (TBD)

sub-option-len: 20 in octets

prefix-length: the length of U_PREFIX64 in bits

U_PREFIX64: the unicast prefix for constructing the IPv4-embedded IPv6 addresses of the multicast sources in SSM mode

Figure 3: DHCPv6 Sub-option Format for U_PREFIX64

4. Client Behaviour

To retrieve the IPv6 prefixes to use to synthesize unicast and multicast IPv4-embedded IPv6 addresses, the DHCPv6 client MUST include OPTION_PREFIX64 in its OPTION_ORO.

If the DHCPv6 client receives more than one OPTION_PREFIX64 option from the DHCPv6 server, only the first instance of that option MUST be used.

When OPTION_PREFIX64 option is received from the DHCPv6 server, at most three sub-options MAY be included.

The prefix conveyed in SUB_OPTION_U_PREFIX64 is used to synthesize unicast IPv4-embedded IPv6 addresses as specified in [RFC6052].

The prefix conveyed in SUB_OPTION_M_PREFIX64 is used to synthesize multicast IPv4-embedded IPv6 addresses as specified in [I-D.boucadair-behave-64-multicast-address-format].

5. Server Behaviour

A DHCPv6 server MUST NOT reply with a value for the OPTION_PREFIX64

if the DHCPv6 client has not explicitly included OPTION_PREFIX64 in its OPTION_ORO.

If OPTION_PREFIX64 option is requested by the DHCPv6 client, the DHCPv6 server MUST NOT send more than one OPTION_PREFIX64 option in the response.

One or two SUB_OPTION_M_PREFIX64 sub-options MAY be enclosed in OPTION_PREFIX64 DHCPv6 option. In particular, if only SSM or ASM mode is supported, only one SUB_OPTION_M_PREFIX64 sub-option MUST be returned to the requesting client. If both SSM and ASM mode are supported, two SUB_OPTION_M_PREFIX64 sub-options MUST be returned.

When two SUB_OPTION_M_PREFIX64 sub-options are present, one SUB_OPTION_M_PREFIX64 sub-option MUST convey an IPv6 prefix in SSM range and the other one MUST enclose an IPv6 prefix in the ASM range.

If the IPv6 multicast prefix conveyed in SUB_OPTION_M_PREFIX64 is an SSM prefix, U_PREFIX64 sub-option MUST also be present.

6. Security Considerations

The security considerations in [RFC3315] are to be considered.

7. Acknowledgements

TBD

8. IANA Considerations

A new DHCPv6 option:

OPTION_PREFIX64

and two sub-options:

SUB_OPTION_M_PREFIX64,

SUB_OPTION_U_PREFIX64

need to be assigned by IANA.

9. References

9.1. 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-03 (work in progress), October 2011.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.
- [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.

9.2. Informative References

- [I-D.ietf-softwire-dslite-multicast]
Wang, Q., Qin, J., Boucadair, M., Jacquenet, C., and Y. Lee, "Multicast Extensions to DS-Lite Technique in Broadband Deployments", draft-ietf-softwire-dslite-multicast-00 (work in progress), September 2011.
- [I-D.ietf-softwire-mesh-multicast]
Xu, M., Cui, Y., Yang, S., Wu, J., Metz, C., and G. Shepherd, "Softwire Mesh Multicast", draft-ietf-softwire-mesh-multicast-01 (work in progress), October 2011.
- [I-D.venaas-behave-mcast46]
Venaas, S., Asaeda, H., SUZUKI, S., and T. Fujisaki, "An IPv4 - IPv6 multicast translator", draft-venaas-behave-mcast46-02 (work in progress), December 2010.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.
- [RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the

IPv4 Address Shortage", RFC 6346, August 2011.

Authors' Addresses

Jacni Qin
ZTE
Shanghai,
China

Phone: +86 1391 8619 913
Email: jacni@jacni.com

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Phone:
Email: mohamed.boucadair@orange.com

Tina Tsou
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +1 408 330 4424
Email: tina.tsou.zouting@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2012

X. Li
C. Bao
CERNET Center/Tsinghua
University
W. Dec
R. Asati
Cisco Systems
C. Xie
Q. Sun
China Telecom
October 31, 2011

dIVI-pd: Dual-Stateless IPv4/IPv6 Translation with Prefix Delegation
draft-xli-software-divi-pd-01

Abstract

This document presents the address specifications and deployment considerations of address-sharing dual stateless IPv4/IPv6 translation with prefix delegation (dIVI-pd). The dIVI-pd keeps the features of stateless, end-to-end address transparency and bidirectional-initiated communications of the original stateless IPv4/IPv6 translation, while it can utilize the IPv4 addresses more effectively. In addition, it does not require the DNS64 and ALG, and can be used with prefix delegation.

Status of this Memo

This Internet-Draft is submitted to IETF 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 May 3, 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	3
2. Applicability	3
3. Terminologies	5
4. Port Mapping Algorithm and Address Format	5
4.1. Port Mapping Algorithm	5
4.2. Basic Mapping Rule (BMR)	6
4.3. Default Mapping Rule (DMR)	7
4.4. Address Specifications	7
5. Header Translation and MTU Handling	7
6. Dual Stateless Translation	8
7. Deployment Considerations	9
8. CE Configuration via DHCP Option	9
9. Experimental Evaluation	10
10. Security Considerations	10
11. IANA Considerations	10
12. Acknowledgments	10
13. References	11
13.1. Normative References	11
13.2. Informative References	12
Authors' Addresses	13

1. Introduction

The experiences for the IPv6 deployment in the past 10 years strongly indicate that for a successful transition, the communication between IPv4 and IPv6 address families should be supported.

Recently, the stateless and stateful IPv4/IPv6 translation methods are developed and became the IETF standards. The original stateless IPv4/IPv6 translation (stateless 1:1 IVI) is scalable, maintains the end-to-end address transparency and support both IPv6 initiated and IPv4 initiated communications [RFC6052], [RFC6144], [RFC6145], [RFC6147], [RFC6219]. But it can not use the IPv4 addresses effectively. The stateful IPv4/IPv6 translation can share the IPv4 addresses among IPv6 hosts, but it only supports IPv6 initiated communication [RFC6052], [RFC6144], [RFC6145], [RFC6146], [RFC6147]. In addition, both stateless and stateful IPv4/IPv6 translation technologies require the application layer gateway (ALG) for the applications which embed IP address literals. Furthermore, in ADSL and 3G environment, it requires the prefix delegation (assigning an IPv6 /64 or shorter) to the customer router/L3-device rather than assigning a single IPv4-translatable address to the customer device defined in [RFC6052].

In this document, we present address specifications and deployment considerations for address-sharing dual stateless IPv4/IPv6 translation with prefix delegation (dIVI-pd), which is based on basic dIVI model [I-D.xli-behave-divi] with the support of prefix delegation. The dIVI-pd can solve the IPv4 address sharing, the ALG and prefix delegation problems mentioned above, though still keeps the stateless, end-to-end address transparency and supporting of both IPv6 initiated and IPv4 initiated communications.

Due to the introduction of the second translation and the prefix delegation, the dIVI-PD is 4-6-4 model and there is a strong correlation to the stateless encapsulation approach [I-D.murakami-software-4rd]. This document uses the address format, the port mapping algorithm and DHCP options defined in [I-D.mdt-software-mapping-address-and-port]. [I-D.mdt-software-map-dhcp-option], which are the joint design works of stateless encapsulation and dual stateless translation.

2. Applicability

The address-sharing dual stateless IPv4/IPv6 translation with prefix delegation (dIVI-pd) can be used in ADSL or 3G environment when prefix delegation is required. An ADSL example is shown in the following figure.

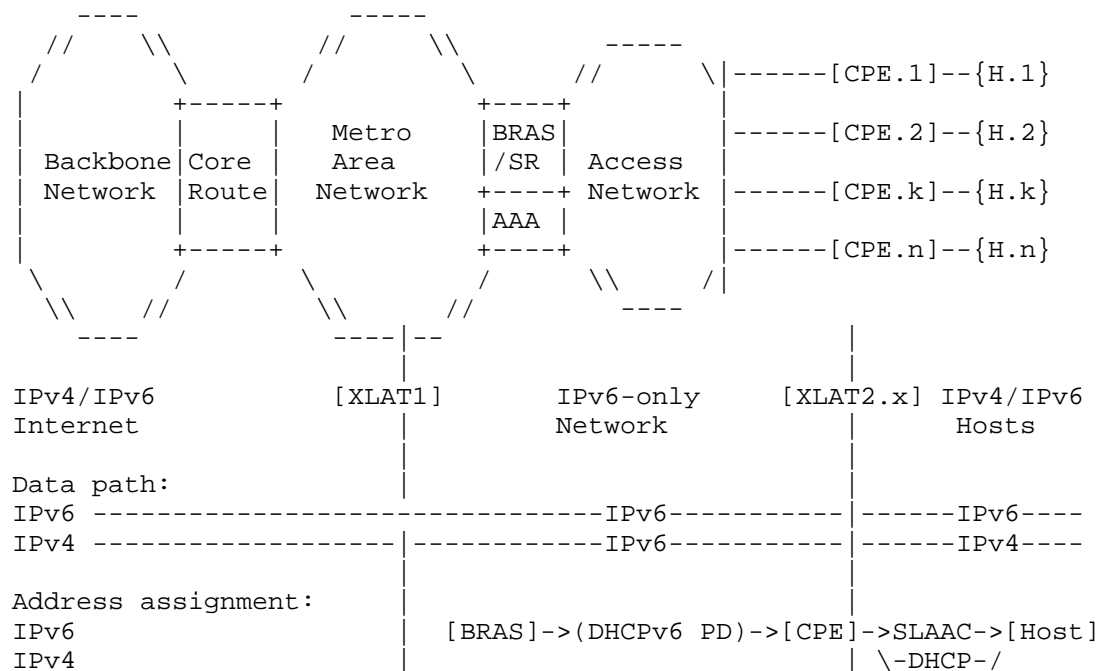


Figure 1: BRAS

Where the ISP's backbone network is dual stack, as well as part of the metro-area network. The core IPv4/IPv6 translator (XLAT1) is performing the IPv4 address-sharing stateless IPv4/IPv6 translation and connects the dual-stack part and the IPv6-only part of the metro-area networks. The access network is IPv6-only and multiple IPv4/IPv6 translators (XLAT2.x) are connected to the access network and provide dual-stack access to the customer devices. Each dual-stack customer get a whole IPv6 /64 (or shorter) and a fractional public IPv4 address.

The data path of this user case are: The IPv6 packets from customer devices and the IPv6 Internet are not translated, while the IPv4 packets from customer devices and the IPv4 Internet are translated twice via stateless IPv4/IPv6 translation technology. Due to the stateless nature, the dual stateless IPv4/IPv6 translation is almost equivalent to tunneling with header compression.

There are two address assignment processes: (1) From BRAS to CPE is via IPv6CP and DHCPv6 prefix delegation; (2) From CPE to customer device, the IPv6 is via SLAAC and the IPv4 is via DHCP. Note that if more than one customer device requires IPv4 addresses, a built-in NAT44 in each CPE can be used to translate a fractional IPv4 address

to several [RFC1918] defined IPv4 addresses.

3. Terminologies

This document uses the terminologies defined in [I-D.mdt-softwire-mapping-address-and-port].

This document uses the terminologies defined in [RFC6144].

Since [I-D.mdt-softwire-mapping-address-and-port] is used for both encapsulation and stateless translation, the equivalent terminologies in [RFC6144] are:

MAP Border Relay (BR) Address: The MAP Border Relay (BR) Address is the IPv4-converted address defined in [RFC6144] and in [RFC6052].

MAP Customer Edge (CE) Address: The MAP Customer Edge (CE) Address is the IPv4-translatable address defined in [RFC6144] and in [RFC6052].

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

4. Port Mapping Algorithm and Address Format

The port mapping algorithm and address format are defined in [I-D.mdt-softwire-mapping-address-and-port].

4.1. Port Mapping Algorithm

Port mapping algorithm is defined in Section 4.1 of [I-D.mdt-softwire-mapping-address-and-port].

For given sharing ratio (R) and the maximum number of continue ports (M), the generalized modulus algorithm is defined as

1. The port number (P) of a given PSID (K) is composed of

$$P = R * M * j + M * K + i$$

Where

- o PSID: K=0 to R-1
- o Port range index: j = (1024/M)/R to ((65536/M)/R)-1, if the well-known port numbers (0-1023) are excluded.
- o Port continue index: i=0 to M-1

2. The PSID (K) of a given port number (P) is determined by

$$K = (\text{floor}(P/M)) \% R$$

Where

o % is modular operator

o floor(arg) is a function returns the largest integer not greater than arg

4.2. Basic Mapping Rule (BMR)

Basic mapping rule is used for IPv4 prefix, address or port set assignment.

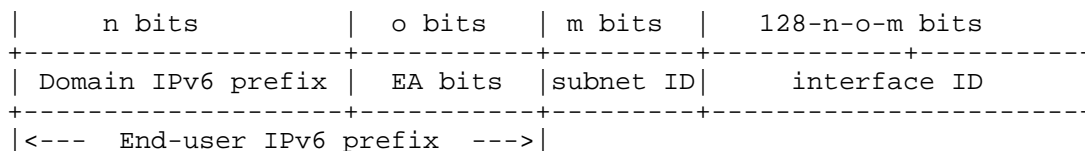


Figure 2: IPv6 address format

The Embedded Address bits (EA bits) are unique per end user within a Domain IPv6 prefix. The EA bits encode the CE specific IPv4 address and port information. The EA bits can contain a full or part of an IPv4 prefix or address, and in the shared IPv4 address case contains a Port Set Identifier (PSID).

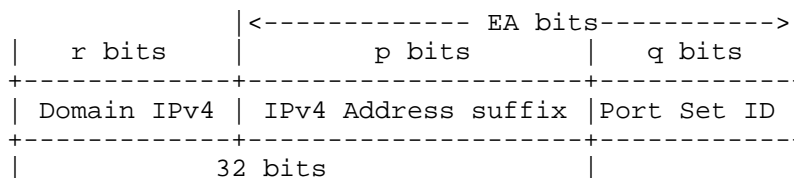


Figure 3: Shared IPv4 address

The interface ID is defined as

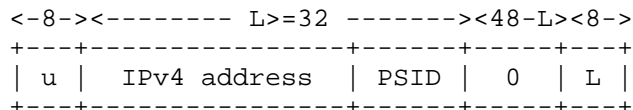


Figure 4: Interface ID

Forwarding Mapping Rule (FMR) is used for forwarding, which has similar address format to BMR. Specifying forwarding mapping rule determines the prefix of the IPv4-translatable addresses for other CEs, which results in different routing behaviors (Hubs and Spokes or Mesh).

4.3. Default Mapping Rule (DMR)

The Default mapping rule defines an IPv6 prefix (BR's IPv6 prefix). The full destination IPv4 address must be encoded in the IPv6 address.

4.4. Address Specifications

Based on the above discussion, the addresses are defined in the following figure.

Source address from a CE to any destination
(IPv4-translatable address)

64		>8		><		L>=32		<>-44-L-<>8		<
Domain prefix	EA bits	0	u	IPv4 address	PSID	0		L		

Destination address from a CE to the outside IPv4 Internet
(IPv4-converted address)

64		>< 8 ><		32		><		24		>	
BR prefix				u	IPv4 address				0		

Figure 5: Extended IPv4-translatable address format

5. Header Translation and MTU Handling

The general header and ICMP translation specifications are defined in [RFC6145].

Special MTU and fragmentation actions must be taken in the case of dual translation.

6. Dual Stateless Translation

When dual stateless IPv4/IPv6 translation is deployed, its behavior is similar to tunneling. Tunneling do not require DNS64 and ALG., because the communication occurs in same address family. Dual translation don't need DNS64 and ALG as well, even in each translator the communication occurs between different address families. However, there are following differences:

- o Scalability. Dual stateless translation is based on routing, there is nothing needed to maintain in the translator, operator's management loads are minimum compared with tunneling scheme, which has to maintain tunnel states.
- o Low OPEX. Dual stateless translation can do traffic engineering and flow analysis without decapsulation which is a must in tunnel case.
- o Header Compression. The dual stateless IPv4/IPv6 translation does not need to do encapsulation and 12 octets header overhead are reduced.
- o Transparent transition to IPv6. The dual stateless translation can be treated as a special case of single stateless translation, the first XLAT performances exactly the same function, no matter there is a XLAT.x or not. Hence it is a unified approach, rather than special setup for the coexistence and transition. This is to say that the ISP can deploy IPv6-only network with XLAT, so the IPv6-only hosts can communicate with both the IPv6 Internet and the IPv4 Internet. However, if for some reason a specific ALG cannot be supported, and for users, who need that specific application, can deploy XLAT.x. When the application is updated, the XLAT.x can be removed. There is nothing to change to XLAT. with more and more contents and users move to IPv6, the working load of XLAT will be less and less, and eventually can be removed. The whole process is transparent, smooth and incremental.

Due to the differences between the IPv4 header and the IPv6 header, the dual stateless IPv4/IPv6 translation cannot be entirely lossless [RFC6145], for example the IPv4 options are lost. The experimental data shows that the IPv4 packets which contain options are very few ($10e-6$) and causes no harm. Another corner case is the fragmentation handling. For IPv4 packets with DF=1 and MF=1, the dual stateless translation will results in DF=0. The experimental data shows that the IPv4 packets with DF=1 and MF=1 are very few ($10e-5$) and causes no harm.

Note that for dual stateless translation, the encapsulation (from

IPv4 to IPv6) and decapsulation (from IPv6 to IPv4) defined by [RFC2473] can be implemented in the translators. In this case, the dual stateless translation processes are entirely lossless, it still has the operation and management conveniences of the dual stateless translation in layer 3, but the control in layer 4 is lost.

7. Deployment Considerations

Given:

1. The total number of CEs in this domain.
2. The sharing ratio R .
3. The port continue parameter M .
4. The customer prefix length.
5. The ISP's IPv6 prefix.
6. The ISP's IPv4 prefix.
7. The BR IPv6 prefix.

Other dIVI-PD configuration parameters can be derived using the port mapping algorithm and address format defined in this document.

8. CE Configuration via DHCP Option

Based on the address format and the port mapping algorithm defined in this document, the CE needs to get the corresponding parameters via DHCPv6 [RFC3315][RFC3633] or others signaling scheme. These parameters are:

1. The IPv6 prefix
2. The IPv6 prefix length
3. The IPv4 prefix
4. The IPv4 prefix length
5. The sharing ratio (R)
6. The maximum number of continue ports (M)

7. The PSID (K)

8. The PSID length (c)

9. Experimental Evaluation

The basic stateless IPv4/IPv6 translation (IVI) has been deployed since 2007. It connects [CERNET] and [CNGI-CERNET2].

The dual stateless translation with IPv4 address sharing (dIVI) has been deployed in [CERNET] and [CNGI-CERNET2] since 2009. The design and implementation results are presented in [I-D.xli-behave-divi].

The dIVI has also been tested in China Telecom. The [I-D.sunq-v6ops-ivi-sp] summarizes the testing results.

The dIVI-pd presented in this document has been running in [CERNET] and [CNGI-CERNET2] since Jan. 2011. The experimental results indicate that the CPE index coding, the suffix coding and port-set ID mapping algorithm work for existing applications without any problem.

10. Security Considerations

See security considerations presented in [RFC6052] and [RFC6145].

11. IANA Considerations

This memo adds no new IANA considerations.

Note to RFC Editor: This section will have served its purpose if it correctly tells IANA that no new assignments or registries are required, or if those assignments or registries are created during the RFC publication process. From the author's perspective, it may therefore be removed upon publication as an RFC at the RFC Editor's discretion.

12. Acknowledgments

The authors would like to acknowledge the following contributors in the different phases of the address-sharing IVI and dIVI development: Hong Zhang, Yu Zhai, Wentao Shang, Weifeng Jiang, Bizhen Fu, Guoliang Han and Weicai Wang.

The authors would like to acknowledge the following contributors who

provided helpful inputs: Heyu Wang, Lu Yan, Dan Wing, Fred Baker, Dave Thaler, Randy Bush, Kevin Yin and Bobby Li.

The authors would like to thank the MAP team for the technical discussions, which make the continue improvements of dIVI-PD.

13. References

13.1. 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.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.
- [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.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

- [RFC6219] Li, X., Bao, C., Chen, M., Zhang, H., and J. Wu, "The China Education and Research Network (CERNET) IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition", RFC 6219, May 2011.
- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

13.2. Informative References

- [CERNET] "CERNET Homepage:
http://www.edu.cn/english_1369/index.shtml".
- [CNGI-CERNET2]
"CNGI-CERNET2 Homepage:
http://www.cernet2.edu.cn/index_en.htm".
- [I-D.bcx-behave-address-fmt-extension]
Bao, C. and X. Li, "Extended IPv6 Addressing for Encoding Port Range", draft-bcx-behave-address-fmt-extension-01 (work in progress), October 2011.
- [I-D.mdt-softwire-map-dhcp-option]
Mrugalski, T., Boucadair, M., and O. Troan, "DHCPv6 Options for Mapping of Address and Port", draft-mdt-softwire-map-dhcp-option-00 (work in progress), October 2011.
- [I-D.mdt-softwire-mapping-address-and-port]
Troan, O., "Mapping of Address and Port (MAP)", draft-mdt-softwire-mapping-address-and-port-00 (work in progress), October 2011.
- [I-D.murakami-softwire-4rd]
Murakami, T., Troan, O., and S. Matsushima, "IPv4 Residual Deployment on IPv6 infrastructure - protocol specification", draft-murakami-softwire-4rd-01 (work in progress), September 2011.
- [I-D.sunq-v6ops-ivi-sp]
Sun, Q., Xie, C., Li, X., Bao, C., and M. Feng, "Considerations for Stateless Translation (IVI/dIVI) in Large SP Network", draft-sunq-v6ops-ivi-sp-02 (work in progress), March 2011.
- [I-D.xli-behave-divi]
Bao, C., Li, X., Zhai, Y., and W. Shang, "dIVI: Dual-Stateless IPv4/IPv6 Translation", draft-xli-behave-divi-04

(work in progress), October 2011.

[RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.

Authors' Addresses

Xing Li
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN

Phone: +86 10-62785983 begin_of_the_skype_highlighting +86 10-62785
983 end_of_the_skype_highlighting
Email: xing@cernet.edu.cn

Congxiao Bao
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN

Phone: +86 10-62785983 begin_of_the_skype_highlighting +86 10-62785
983 end_of_the_skype_highlighting
Email: congxiao@cernet.edu.cn

Wojciech Dec
Cisco Systems
Haarlerberdweg 13-19
Amsterdam 1101 CH
NL

Email: wdec@cisco.com

Rajiv Asati
Cisco Systems
7025-6 Kit Creek Road
Research Triangle Park NC 27709
USA

Email: rajiva@cisco.com

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

Phone: +86-10-58552116 begin_of_the_skype_highlighting +86-10-58552
116 end_of_the_skype_highlighting
Email: xiechf@ctbri.com.cn

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

Phone: +86-10-58552936 begin_of_the_skype_highlighting +86-10-58552
936 end_of_the_skype_highlighting
Email: sunqiong@ctbri.com.cn

SAVI
Internet Draft
Intended status: Standard Tracks
Expires: May 2012

K.Xu, G.Hu, J.Bi, M.Xu
Tsinghua Univ.
F.Shi
China Telecom
November 20, 2011

The Requirements and Tentative Solutions for SAVI in IPv4/IPv6
Transition
draft-xu-savi-transition-00.txt

Abstract

SAVI Working Group is developing standardize mechanisms that prevent nodes attached to the same IP link from spoofing each other's IP addresses, and achieve IP source address validation at a finer granularity. Unfortunately, up to now, SAVI switch only works under the scenario of pure wire/wireless IPv6 Ethernet access subnet. In the current stage of IPv4/IPv6 transition which can't be cross over, SAVI has to make more progress to adapt it. This document describes the requirements and gives tentative solutions for the SAVI in IP4/IPv6 transition period. In RFC5565, Wu et.al proposal a softwire mesh framework to address the problem of routing information and data packets of one protocol how to pass through a single-protocol network of the other protocol. According to the real situation of CNGI-CERNET and China Telecom, document takes scenario of IPv4 packets transit IPv6 network into account.

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 May 20, 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	3
2. Conventions used in this document.....	4
3. Requirements and solutions for SAVI in IPv4/IPv6 transition..	4
3.1. Public 4over6	4
3.2. Lightweight 4over6.....	6
4. Conclusions	6
5. References	6
5.1. Normative References.....	6
6. Acknowledgments	6

1. Introduction

Without a doubt, SAVI has made significant contribution for IP source address validation and anti-spoofing in the scenario of pure IPv6 Ethernet access subnet. Current situation is that IPv4 address has worn out but still takes a domination position for a long time, and IPv6 networking start to thrive. Meanwhile, SAVI switch only works at the scenarios of wire or wireless Ethernet, thus, there are lots of works have to do and many efforts need to make for adapting with the reality and promoting SAVI scheme. In the transition period from IPv4 to IPv6, approaches are classified into three types: dual stack, tunneling and translation. Regarding to real situation of CNGI-CERNET and China Telecomm which are the two of biggest Internet providers, this document mainly states the requirements and proposes some tentative solutions for scenarios of public 4over6[p4over6], lightweight 4over6[l4over6], which are the two implementations for scenario of IPv4-over-IPv6 in RFC5565. We hope that our proposal would be helpful for resolving problems of IP spoofing and validation in users' access subnet under transition period.

Public IPv4 over Access IPv6 Network (4over6) is a mechanism for bidirectional IPv4 communication between IPv4 Internet and end hosts or IPv4 networks sited in IPv6 access network. This mechanism follows the software 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.

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. In the host initiator case, every host will get one IPv4 address; in the CPE (Customer premises equipment) case, every CPE will get one IPv4 address, which will be shared by hosts behind the CPE.

There is a slight different between public 4over6 and lightweight 4over6. Briefly, lightweight 4over6 mitigates IPv4 address exhaustion by sharing public IPv4 addresses amongst users, while public 4over6 host own a unique public IPv4 address. In lightweight 4over6 scenario, several hosts share a public address but have different port range by extending DHCPv4 and PCP protocol.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Requirements and solutions for SAVI in IPv4/IPv6 transition

In this section, we mainly talk about the requirements for SAVI in transition period regarding to public 4over6 and IVI approaches.

3.1. Public 4over6

Figure 1 illustrates the working scenario of public 4over6. Users in an IPv6 network take IPv6 as their native service. There are two types of users: dual-stack and CPE behind. 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".

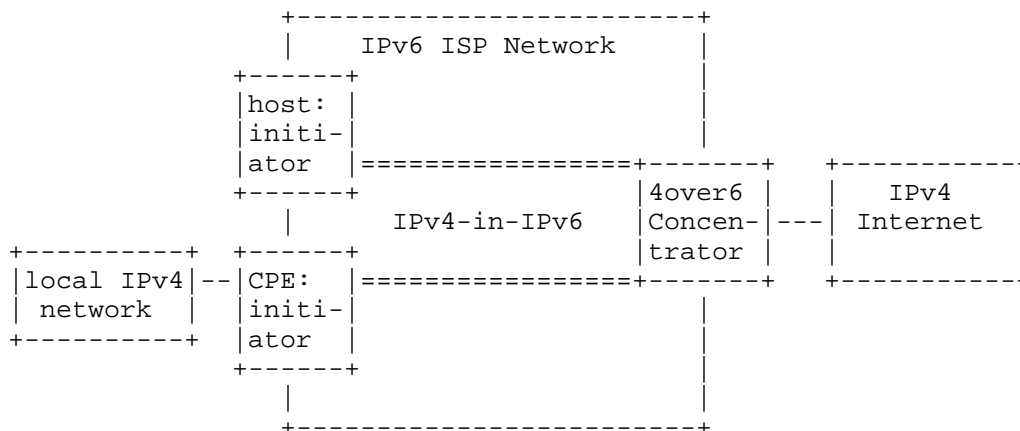


Figure 1 Public 4over6 scenario

Public 4over6 has stateful and stateless two working mode. Either stateful or stateless mode depends on whether it needs a mapping record for IPv4 and its corresponding IPv6 address in 4over6 Concentrator or not. The difference among them is that stateless mode

use the IPv6 address based on IPv4 embedded and initiator and concentrator both needs to parse/compose the address, while stateful mode means concentrator should restore the mapping records for IPv4/IPv6 address. Two types of users use DHCP or PCP protocol to retrieve IP address.

Two types of users multiple two types of working modes, that is four scenarios, we analyse them in detail.

a) Scenario 1: Dual-stack with stateful

For accessing IPv4 and IPv6 resources, this type of users owns IPv4 and IPv4 unrelated IPv6 addresses. Dual-stack hosts get their IPv6 address via DHCPv6 protocol as normal, however, IPv4 addresses allocation datagram for them need to encapsulate into tunnel, tunnel initiator is their IPv6 addresses and the 4over6 concentrator is the end of tunnel. For the reason of the toughness in access switch to parse IPv4 address from tunnel, SAVI switch only needs to snoop DHCPv6/PCP protocols and bind the relationship of <IPv6, MAC, Switch-Port>, however, 4over6 concentrator validates the mapping relationship of IPv4 to IPv6.

b) Scenario 2: Dual-stack with stateless

Even though this type of users has both IPv4 and IPv6 address, however, any of them could conduct to another. SAVI switch saves the relationship of <IPv6, MAC, Switch-Port > or <IPv4, IPv6, MAC, Switch-Port> would be ok.

c) Scenario 3: CPE-behind with stateful

In this scenario, hosts only have public IPv4 address and CPE plays the role of broker with dual-stack. SAVI switch should to snooping the DHCPv4/PCP protocols interaction and bind <IPv4, MAC, Switch-Port> relationship.

d) Scenario 4: CPE-behind with stateless

SAVI switch does the same thing with scenario C, the difference is that the start point of tunnel is initiated by CPE which use an IPv4-mapped IPv6 address.

In summary, SAVI switch should listen to the IP address allocation protocol like DHCPv6, DHCPv4, PCP etc. and bind host's properties based on working scenario.

3.2. Lightweight 4over6

We no longer carry out a detailed description of each scenario in lightweight 4 over6 because there is nothing big changes compare with public 4over6 scheme. The difference exists in the way of host how to own an address. As mentioned before, public 4over6 host entirely own a unique public IPv4 address, while several lightweight 4over6 hosts share a public IPv4 address, but they have different port range. This change needs to extend DHCPv4[DHCPv6-map] and PCP protocol, thus, SAVI switch needs to listen to these address allocation protocols and bind relationship of <IPv4, MAC, Switch-Port, Port-range>.

4. Conclusions

There would be a long period from IPv4 to IPv6, public 4over6 is one of practical approach for inter-communication in the transition stage. SAVI switch focus on anti-snooping in users' access subnet by binding hosts' information. But till now, it only works at IPv6 environment. This document presents the SAVI requirements in this period, and in the meanwhile, we investigated working scenarios of public 4over6 in detail and gave some tentative solutions for SAVI adaption.

5. References

5.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5565] J.Wu, Y.Cui, C.Metz, E.Rosen, "'Software Mesh Framework'", RFC 5565, June 2009.
- [p4over6] Y.Cui, J.Wu, P.Wu, C.Metz, O.Vautrin, Y.Lee, "Public IPv4 over Access IPv6 Network draft-cui-software-host-4over6-06", Internet-Draft, July 2011
- [l4over6] Y.Cui, J.Wu, P.Wu, Q. Sun, C. Xie, C. Zhou, Y.Lee, "Lightweight 4over6 in access network draft-cui-software-b4-translated-ds-lite-04", Internet-Draft, Oct. 2011
- [DHCPv6-map] T. Mrugalski, M. Boucadair, O. Troan, X. Deng, C. Bao, "DHCPv6 Options for Mapping of Address and Port draft-mdt-software-map-dhcp-option-01'", Internet-Draft, Oct. 2011

6. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Ke Xu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing, 100084
China
Email: xuke@mail.tsinghua.edu.cn

Guangwu Hu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
China
EMail: hgw09@mails.tsinghua.edu.cn

Fan Shi
China Telecom
Beijing Research Institute, China Telecom
Beijing 100035
China
EMail: shifan@ctbri.com.cn

Jun Bi
Tsinghua University
Network Research Center, Tsinghua University
Beijing 100084
China
Email: junbi@tsinghua.edu.cn

Mingwei Xu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
China
Email: xmw@csnet1.cs.tsinghua.edu.cn

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: May 2012

Xiaohong Deng
M. Boucadair
France Telecom
C. Zhou
Huawei Technologies
October 31, 2011

NAT offload extension to Dual-Stack lite
draft-zhou-softwire-b4-nat-04

Abstract

Dual-Stack Lite, combining IPv4-in-IPv6 tunnel and Carrier Grade NAT technologies, provides an approach that offers IPv4 service via IPv6 network by sharing IPv4 addresses among customers during IPv6 transition period. Dual-stack lite, however, requires CGN to maintain active NAT sessions, which means processing performance, memory size and log abilities for NAT sessions should scale with number of sessions of subscribers; Hence increasing in CAPEX for operators would be resulted in when traffic increase.

This document propose the NAT offload extensions to DS-Lite, which allows offloading NAT translation function from centralized network side (AFTR) to distributed customer equipments (B4), thereby offering a trade-off between CAPEX (e.g. less performance requirements on AFTR device) and OPEX (e.g., easy and fast deployment of Dual-Stack Lite) for operators. The ability of easily co-deploying with basic Dual-Stack Lite is essential to NAT offload extension to DS-Lite.

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."

DBC

Expires May 31, 2012

[Page 1]

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. Background.....	3
2. NAT offload extended DS-Lite Overview and terminologies.....	3
3. NATed B4 Behavior.....	5
3.1. Plain IPv4 Address.....	5
3.2. Restricted IPv4 Address and port set provisioning.....	5
3.2.1. Restricted port allocation strategies and requirements..	5
3.2.2. Restricted IPv4 Address and port set provisioning method	
.....	6
3.3. Outgoing Packets Processing.....	6
3.4. Incoming Packets Processing.....	6
3.4.1. Incoming Ports considerations on a given restricted IPv4	
address.....	6
4. NAT offload AFTR Behaviour.....	7
4.1. Outgoing Packets Processing.....	7
4.2. Incoming Packets Processing.....	7
5. Fragmentation and Reassembly and DNS.....	7
6. Security Considerations.....	8
7. IANA Considerations.....	10
8. References.....	10
8.1. Normative References.....	10
8.2. Informative References.....	10
9. Acknowledgments.....	12

1. Background

The basic idea of NAT offload extension to DS-lite, is to reuse the basic DS-Lite infrastructure, including tunneling transport and provisioning method, and ICMP and fragmentation processing as well.

The NAT offload extension makes the AFTR table scales with customer number other than traffic sessions. Based on this NAT offload extension, log entries for per subscriber instead of per session is achievable. IPv4 address utilization efficiency depends on port allocation strategies, e.g., per port on demand, or a buck of ports pre-allocation, which would be elaborated in Section 5.

Besides, this method allows unique IPv6 address for delivery both IPv4 over IPv6 traffic and native IPv6 traffic without introduce any IPv4 addressing/routing into IPv6 address/routing, as it reuses Dual Stack Lite tunneling transport infrastructure, unlike stateless solutions with port set allocation such as aplusp and 4rd, that either requires two IPv6 addresses separately for either IPv4 traffic over IPv6 or native IPv6 traffic, or require carefully design to avoid introduce IPv4 routing to IPv6 routing when using unique IPv6 address to transport both IPv4 over IPv6 traffic and native IPv6 traffic.

2. NAT offload extended DS-Lite Overview and terminologies

Figure 1 provides an overview of the NAT offload extended DS-Lite.

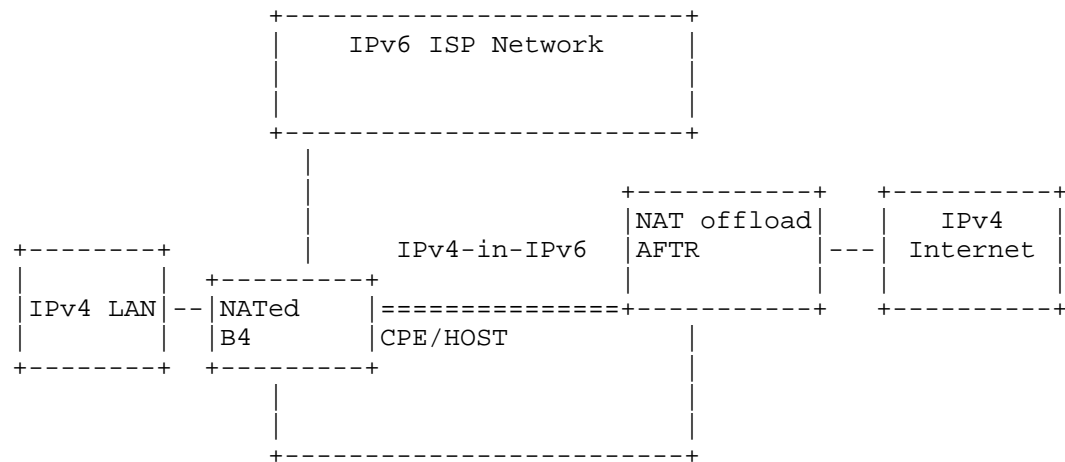


Figure 1 : NAT offload extended DS-Lite Overview

NATed B4: A NAT offload extended B4 which is called NATed B4 in this document can be either an IPv6 hosts or a CPE. NATed B4 performs IP address and port translation function, besides establishment of IPv4 in IPv6 tunnel with AFTR.

NAT offload AFTR: A NAT offload extended AFTR which is called NAT offload AFTR is responsible for establishing IPv4 in IPv6 tunneling with NATed B4 to transport IPv4 over IPv6 while the NAT translation function is offloaded to NATed B4.

A NATed B4 uses IPv4 address with a restricted port set for this IPv4 connectivity, which may be provisioned via either DHCPv4 with the AFTR, or via PCP with the PCP server. The AFTR keeps the mapping between B4's IPv6 address, allocated IPv4 address, and a restricted port set ID on a per customer basis.

For host NATed B4 case, the host gets public address directly. It is also suggested that the host run a local NAT to map randomly generated ports into the restricted port set. Private to public address translation would not be needed in this NAT. Another

Internet-Draft Lightweight extension to DS-Lite October 2011
solution is to have the IP stack to only assign ports within the
restricted port set to applications. Either way the host guarantees
that every port number in the packets sent out by itself falls into
the allocated port set.

3. NATed B4 Behavior

The NATed B4 is responsible for performing NAT and/ALG functions,
basic B4 functions, as well as supporting NAT Traversal mechanisms
(e.g., UPnP or NAT-PMP).

The tunneling provisioning of the B4 element should reuse what has
defined in [I-D.ietf-softwire-dual-stack-lite].

3.1. Plain IPv4 Address

A NATed 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].

3.2. Restricted IPv4 Address and port set provisioning

3.2.1. Restricted port allocation strategies and requirements

Restricted port allocation strategies for this approach could either
be allocating per port on demand, or be pre-allocating a port set (no
matter a continuous port range, or multiple non-continuous sub port
sets), which leads to trade-off between provisioning efficiency and
IPv4 utilization efficiency.

Note that efficiency on log is reported by operators as a practical
requirement for AFTR, hence port set decoding should take this
requirement into account, no matter which port allocation strategy is
adopt.

Internet-Draft Lightweight extension to DS-Lite October 2011
Unlike stateless 4over6 solutions such as [I-D.murakami-softwire-4rd], the restricted port sets allocation for NAT offload extended DS-Lite has no requires on careful planning of the IPv6 and IPv4 addressing together. It therefore offers more flexibility for ISPs, when it comes to managing the IPv6 access network, and introduces no impact on IPv6 routing.

3.2.2. Restricted IPv4 Address and port set provisioning method

Either DHCP for example, [I-D.bajko-pripaddrassign] or PCP would be candidate for delivery Restricted IPv4 and port set.

With PCP, The basic PCP protocol allows per port on demand allocation, while an extension to PCP [I-D.tsou-pcp-natcoord] supports pre-allocate bulk of ports.

3.3. 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.

3.4. 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.

3.4.1. Incoming Ports considerations 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

Internet-Draft Lightweight extension to DS-Lite October 2011
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.

4. NAT offload AFTR Behaviour

The NAT offload AFTR may be co-located with IP and /or restricted
port set allocation server (e.g., a DHCP server, or a PCP server).

The AFTR only maintains a static mapping entry per customer consist
of IPv6 address, IPv4 address and port set ID, other than maintains
NAT entries per session.

4.1. Outgoing Packets Processing

For outgoing packets, the NAT offload AFTR simply decapsulates it and
forwards it to IPv4 Internet.

4.2. Incoming Packets Processing

For inbound traffic, NAT offload AFTR would use the IPv4 destination
address and port as the index to retrieve mapping table in order to
find a destination IPv6 address, and then encapsulates it into IPv6,
so that native IPv6 routing could be used to forward the IPv4 in IPv6
traffic.

5. Fragmentation and Reassembly and DNS

No change to Section 5.3 of [I-D.ietf-softwire-dual-stack-lite]. The
DNS behavior is the same as described in [I-D.ietf-softwire-dual-
stack-lite].

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.

In this section, a non-continuous restricted port set encoding/decoding and an algorithm of random ephemeral port selection within the allocated restricted port set example proves that port randomization is applicable this approach.

On every external IPv4 address, according to port set size N , $\log_2(N)$ bits are randomly choosing by NAT offload AFTR as subscribers identification bits (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 bits.

1st	2nd	3rd	4th	5th	6th	7th	8th
0	s	0	0	s	0	s	0
9th	10th	11th	12th	13th	14th	15th	16th
s	0	s	0	0	0	0	0

Figure 2 : 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. The NAT offload AFTR can use subscriber ID pattern as port set ID on a per restricted IPv4 address basis, which allows log entries scale on a subscriber basis, hence meets the log efficiency requirements described in Section 3.1.2.

Internet-Draft	Lightweight extension to DS-Lite								October 2011
	1st	2nd	3rd	4th	5th	6th	7th	8th	
	+-----	+-----	+-----	+-----	+-----	+-----	+-----	+-----	
	0	1	0	0	1	0	1	0	
	+-----	+-----	+-----	+-----	+-----	+-----	+-----	+-----	
	9th	10th	11th	12th	13th	14th	15th	16th	
	+-----	+-----	+-----	+-----	+-----	+-----	+-----	+-----	
	1	0	1	0	0	0	0	0	
	+-----	+-----	+-----	+-----	+-----	+-----	+-----	+-----	

Figure 3 : 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.

1st	2nd	3rd	4th	5th	6th	7th	8th	
+-----	+-----	+-----	+-----	+-----	+-----	+-----	+-----	
1	s	1	1	s	1	s	1	
+-----	+-----	+-----	+-----	+-----	+-----	+-----	+-----	
9th	10th	11th	12th	13th	14th	15th	16th	
+-----	+-----	+-----	+-----	+-----	+-----	+-----	+-----	
s	1	s	1	1	1	1	1	
+-----	+-----	+-----	+-----	+-----	+-----	+-----	+-----	

Figure 4 : 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.

```
Internet-Draft      Lightweight extension to DS-Lite      October 2011
do{
    restricted_next_ephemeral = (random()| customer_ID_pattern)
                                & customer_ID_value;
    if(five-tuple is unique)
    return restricted_next_ephemeral;
}
```

Figure 5 : Random ephemeral port selection of restricted port set algorithm.

7. IANA Considerations

TBD.

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.

8.2. Informative 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.

Boucadair, M., Skoberne, N., and W. Dec, "Analysis of
Port Indexing Algorithms",
draft-bsd-software-stateless-port-index-analysis-00
(work in progress), September 2011.

[I-D.cui-software-dhcp-over-tunnel]

Cui, Y., Wu, P., and J. Wu, "DHCPv4 Behavior over IP-IP
tunnel", draft-cui-software-dhcp-over-tunnel-01 (work
in progress), July 2011.

[I-D.cui-software-host-4over6]

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

[I-D.murakami-software-4rd]

Murakami, T., Troan, O., and S. Matsushima, "IPv4
Residual Deployment on IPv6 infrastructure - protocol
specification", draft-murakami-software-4rd-01 (work in
progress), September 2011.

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

9. Acknowledgments

Thank Alain Durand, Ole Troan and Ralph Dorm for their valuable feedback and discussion to this approach, and thanks to Qiong Sun for a discussion from operators needs' perspective.

Appendix A. Variants of this approach

A.1. Introduction

This section defines variants of deployment for this NAT offload DS-Lite approach. A.2 describes its combination with stateless encapsulation.

A.2 Stateless Encapsulation

B4 may implement the stateless encapsulation specified in Section 4.4 of [I-D.ymbk-aplusp].

Xiaohong Deng
France Telecom
Email: xiaohong.deng@orange.com

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.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