

Softwire Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 14, 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  
July 13, 2012

Lightweight 4over6: An Extension to the DS-Lite Architecture  
draft-cui-softwire-b4-translated-ds-lite-07

Abstract

This document specifies an extension to DS-Lite called Lightweight 4over6. This mechanism moves the translation function from the tunnel concentrator (AFTR) to initiators (B4s), and hence reduces the mapping scale on the concentrator to a per-subscriber level. To delegate the NATP function to the initiators, port-restricted IPv4 addresses are allocated to the initiators.

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 14, 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 . . . . .	4
3. Terminology . . . . .	4
4. Lightweight 4over6 Overview . . . . .	5
5. Port-Restricted IPv4 Address Allocation . . . . .	5
6. Lightweight 4over6 Initiator Behavior . . . . .	6
6.1. Initiator Provisioning . . . . .	6
6.2. Initiator Data Plane Behavior . . . . .	7
7. Lightweight 4over6 Concentrator Behavior . . . . .	7
7.1. Binding Table Maintenance . . . . .	7
7.2. Concentrator Data Plane Behavior . . . . .	8
8. Fragmentation and Reassembly . . . . .	9
9. DNS . . . . .	9
10. ICMP Processing . . . . .	9
11. Security Consideration . . . . .	10
12. IANA Considerations . . . . .	10
13. Author List . . . . .	10
14. Acknowledgement . . . . .	12
15. Appendix: Alternatives for Port-Restricted Address Allocation . . . . .	13
16. References . . . . .	13
16.1. Normative References . . . . .	13
16.2. Informative References . . . . .	14
Authors' Addresses . . . . .	15

## 1. Introduction

Dual-Stack Lite (DS-Lite, [RFC6333]) provides IPv4 access over an IPv6 network relying on two functional elements: B4 and AFTR. The B4 element establishes an IPv4-in-IPv6 software to the AFTR and encapsulates IPv4 packets within IPv6 packets. When the AFTR receives these IPv6 packets, it de-capsulates them and then performs NAPT44 [RFC3022] on the IPv4 packets. This procedure allows the AFTR to dynamically assign port numbers to requesting hosts; hence, increasing the port-sharing ratio and utilization (see [RFC6269]). There is a trade-off, however: the AFTR is required to maintain active NAPT sessions. In the centralized deployment model where one AFTR serves a large number of hosts, the huge number of NAPT sessions may become a performance bottleneck. A large NAPT table demands more processing power for maintaining and searching, as well as consumes more memory space. On the other hand, NAPT44 function is already widely supported and used in today's CPE devices. By leveraging this existing NAPT function and perform NATPT44 on the CPEs, the binding table in the centralized AFTR can be significantly reduced, and the AFTR can offload the NAPT functionality.

This document proposes such an extension to the DS-Lite model. The extension is designed to simplify the AFTR element by moving NAPT functionality to the B4 elements. The B4 element is provisioned with an IPv6 prefix, an IPv4 address and a port-set. An IPv6 address from the assigned prefix is used to create the software, while the IPv4 address and port-set is used for NAPT44 in the home gateway (CPE). The CPE performs NAPT on the end user's packets with the IPv4 address and port-set. IPv4 packets are forwarded between the CPE and the AFTR using IPv4-in-IPv6 encapsulation. The AFTR maintains a mapping entry with the CPE's IPv6 address, IPv4 address and port-set per subscriber. For inbound IPv4 packets received by the AFTR, the IPv4 destination address and port are used to find the IPv6 encapsulation destination in the binding table. The AFTR does not maintain any NAPT session entries.

Compared to stateless solutions with port-set allocation such as MAP [I-D.mdt-software-mapping-address-and-port], this mechanism is suitable for operators who prefer to keep IPv6 and IPv4 addressing architectures separated. They can administer native IPv6 network addressing without the influence of IPv4-over-IPv6 requirements. For example, an operator may want to provide IPv4 as an on-demand service in its IPv6 network, based on subscriber requests. The dynamic allocation of IPv4 addresses and port-sets makes more efficient usage of IPv4 resources than stateless solutions in this case.

Another example is: An operator may only have many small and non-contiguous IPv4 blocks available to provide IPv4 over IPv6, rather

than a few large contiguous IPv4 blocks. This mechanism preserves the dynamic feature of IPv4/IPv6 address binding as in DS-Lite, so it does not require the administration and management of many MAP domains in the network and corresponding mapping rules in the CPEs.

The model that is presented here offers a solution for a hub-and-spoke architecture only. It does not offer meshed IPv4 connectivity between subscribers. The simplicity and flexibility of IPv4/v6 address planning and provisioning described here are a tradeoff for this reduced functionality: the subscriber does not need the information of other subscribers.

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 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:

- o Lightweight 4over6: Lightweight 4over6 is an IPv4-over-IPv6 hub and spoke mechanism, which supports address sharing [RFC6269] and performs the IPv4 translation (NAPT44) on the initiator (spoke) side.
- o Lightweight 4over6 initiator (or "initiator"): the tunnel initiator in the Lightweight 4over6 mechanism. The Lightweight 4over6 initiator may be a host directly connected to an IPv6 network, or a dual-stack CPE connecting an IPv4 local network to an IPv6 network. It is collocated with a NAPT44 function in addition to IPv4-in-IPv6 encapsulation and de-capsulation functions.

- o Lightweight 4over6 concentrator (or "concentrator"): the tunnel concentrator in the Lightweight 4over6 mechanism. The Lightweight 4over6 concentrator tunnels IPv4 packets to the IPv4 Internet over an IPv6 network. It provides IPv4-in-IPv6 encapsulation and de-encapsulation functions but does not perform a NAT function.
- o Port-restricted IPv4 address: A public IPv4 address with a restricted port-set. In Lightweight 4over6, multiple initiators may share the same IPv4 address, however, their port-sets must be non-overlapping. Source ports of IPv4 packets sent by the initiator must belong to the assigned port-set.

#### 4. Lightweight 4over6 Overview

Lightweight 4over6 initiators and a Lightweight 4over6 concentrator are connected through an IPv6-enabled network (Figure 1). Both use an IPv4-in-IPv6 encapsulation scheme to deliver IPv4 connectivity services. An initiator uses a port-restricted IPv4 address for IPv4 services delivered over the IPv6-enabled network (See Section 5 for further detail). The concentrator keeps the binding between the initiator's IPv6 address and the allocated IPv4 address + port-set.

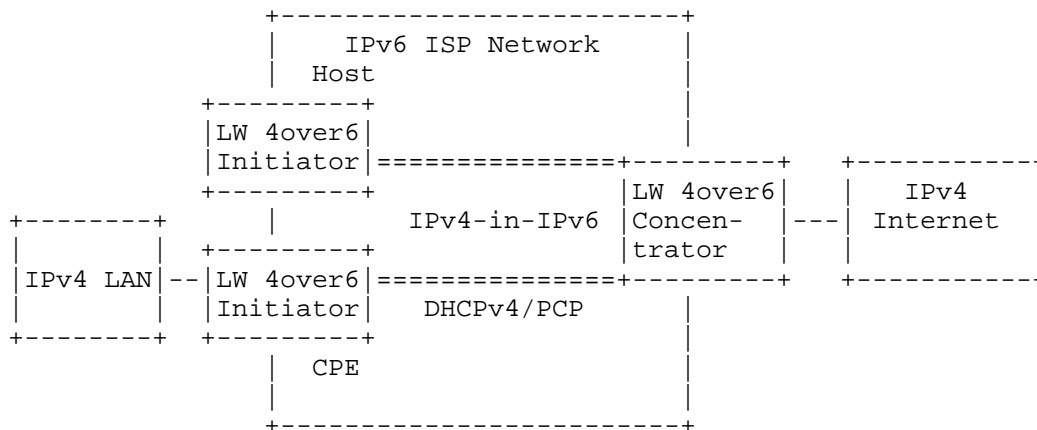


Figure 1 Lightweight 4over6 Overview

#### 5. Port-Restricted IPv4 Address Allocation

In Lightweight 4over6, an initiator is provisioned with a public address and port-set. Different mechanisms can be used for port-restricted IPv4 address provisioning, e.g.- DHCPv4, DHCPv6, PCP, PPP IPCP. The mechanism described in this document uses DHCPv4 as it is

widely deployed in services providers networks and supports all IPv4 and IPv6 addressing models.

DHCPv4 messages between the initiator and the DHCPv4 server MUST be sent over IPv6 [I-D.ietf-dhc-dhcpv4-over-ipv6], and [I-D.bajko-pripaddrassign] MUST be supported for port-set allocation.

Other optional alternatives to retrieve the public address and port-set also exist. The specific protocol extensions are out of scope in this document, however some alternatives are mentioned in the Appendix Section.

## 6. Lightweight 4over6 Initiator Behavior

### 6.1. Initiator Provisioning

To configure the IPv4-in-IPv6 tunnel, the Lightweight 4over6 initiator MUST have the concentrator's IPv6 address. This IPv6 address can be learned through a variety of mechanisms, ranging from an out-of-band mechanism, manual configuration, DHCPv6, etc. In order to guarantee interoperability, a Lightweight 4over6 initiator SHOULD implement the DHCPv6 option defined in [RFC6334]. The initiator MUST use its WAN interface for sourcing the DHCPv6 request as defined in [RFC6333].

Multi-homed CPE devices connected to two or more service providers are not covered as part of this document.

A Lightweight 4over6 initiator MUST support dynamic port-restricted IPv4 address provisioning, by means of implementing the DHCPv4 mechanism (including [I-D.ietf-dhc-dhcpv4-over-ipv6] and [I-D.bajko-pripaddrassign]). The IPv6 address of the DHCPv4 server/relay can be configured using a variety of methods, too, ranging from an out-of-band mechanism, manual configuration, a variety of DHCPv6 options, or taking the concentrator address configuration when collocating with concentrator. In order to guarantee interoperability, an initiator SHOULD implement the DHCPv6 option defined in [I-D.mrugalski-software-dhcpv4-over-v6-option]. If the DHCPv4 over IPv6 client has multiple IPv6 addresses assigned to its WAN interface, the mechanisms defined in RFC3484 MUST be applied for selecting the correct address as the source of the DHCPv4 over IPv6 request. A DHCPv4 over IPv6 client embedded within the initiator MUST use the same IPv6 address as the data plane encapsulation source address for all DHCPv4 over IPv6 requests. In the event the encapsulation source address is changed for any reason (such as the DHCP lease expiring), the DHCPv4 over IPv6 process MUST be re-initiated.

## 6.2. Initiator Data Plane Behavior

The data plane functions of the initiator include address translation (NAPT44), encapsulation and de-capsulation. The initiator runs standard NAPT44 [RFC3022] using the allocated port-restricted address as its external IP and port numbers.

Internally connected hosts source IPv4 packets with an [RFC1918] address. When the initiator 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 concentrator's IPv6 address and the source IPv6 address is the initiator's IPv6 address. Finally, the initiator forwards the encapsulated packet to the configured concentrator.

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

Tunneling MUST be done in accordance with [RFC2473] and [RFC4213].

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

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

## 7. Lightweight 4over6 Concentrator Behavior

### 7.1. Binding Table Maintenance

The Lightweight 4over6 concentrator MUST maintain an address binding table. Each entry in the table contains a public IPv4 address, a port-set and an IPv6 address for a single initiator. The entry has two functions: IPv6 encapsulation of inbound IPv4 packets destined to the initiator and validation of outbound IPv4-in-IPv6 packets received from the initiator for de-capsulation.

The concentrator MUST synchronize the binding information with the port-restricted address provisioning process. With DHCPv4 as the

provisioning method, the initiators send DHCP messages to the DHCP server or relay agent over IPv6. If the concentrator implements a local DHCPv4 server or relay agent, the initiators MAY send the messages to the concentrator; then the concentrator is able to learn the bindings between IPv6 address and IPv4 address with port set directly. If the concentrator 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). The exact mechanism for this is deployment-specific and out of scope. For all provisioning processes, the lifetime of binding table entries MUST be synchronized with the lifetime of address allocations.

## 7.2. Concentrator Data Plane Behavior

The data plane functions of the concentrator are encapsulation and de-capsulation. When the concentrator receives an IPv4-in-IPv6 packet from an initiator, it de-capsulates the IPv6 header and verifies the source addresses and port in the binding table. If the source addresses and port match an entry in the binding table (that is to say, the source IPv6 address in the IPv6 header is identical to the IPv6 address of the entry, the source IPv4 address in the IPv4 header is identical to the IPv4 address of the entry, and the source port falls into the port-set of the entry), the concentrator forwards the packet to the IPv4 destination. If no match is found (e.g., not authorized IPv4 address, port out of range, etc.), the concentrator MUST discard the packet. An ICMP error message MAY be sent back to the requesting initiator. The ICMP policy SHOULD be configurable.

When the concentrator receives an inbound IPv4 packet, it uses the IPv4 destination address and port to lookup the destination initiator's IPv6 address in the binding table. If a match is found, the concentrator encapsulates the IPv4 packet. The source is the concentrator's IPv6 address and the destination is the initiator's IPv6 address from the matched entry. Then, the concentrator forwards the packet to the initiator natively over the IPv6 network. If no match is found, the concentrator MUST discard the packet. An ICMP error message MAY be sent back. The ICMP policy SHOULD be configurable.

Tunneling MUST be done in accordance with [RFC2473] and [RFC4213].

The concentrator MUST support hairpinning of traffic between two initiators, by performing de-capsulation and re-encapsulation of packets.



## 8. Fragmentation and Reassembly

The same considerations as described in Section 5.3 and Section 6.3 of [RFC6333] are to be taken into account.

## 9. DNS

The procedure described in Section 5.5 and Section 6.4 of [RFC6333] is to be followed.

## 10. ICMP Processing

ICMP does not work in an address sharing environment without special handling [RFC6269]. When implementing Lightweight 4over6, the following behaviour SHOULD be implemented to provide basic remote IPv4 service diagnostics for a port restricted CPE: For inbound ICMP messages, the concentrator MAY behave in two modes:

Either:

1. Check the ICMP Type field.
2. If the ICMP type is set to 8 (echo request), then the concentrator MUST discard the packet.
3. If the ICMP field is set to 0 (echo reply), then the concentrator must take the value of the ICMP identifier field as the source port.
4. If the ICMP type field is set to any other value, then the concentrator 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.

Or:

- o Discard all inbound ICMP requests.

The ICMP policy SHOULD be configurable.

The initiator should implement the requirements defined in [RFC5508] for ICMP forwarding. For ICMP echo request packets originating from

the private IPv4 network, the initiator SHOULD implement the method described in [RFC6346] and use an available port from it's port-set as the ICMP Identifier.

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

## 11. Security Consideration

As the port space for a subscriber shrinks significantly due to the address sharing, the randomness for the port numbers of the subscriber is decreased significantly. In other words, 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.

## 12. IANA Considerations

This document does not include any IANA request.

## 13. 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

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

Email: [repenno@cisco.com](mailto:repenno@cisco.com)

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

Email: [axel.clauberg@telekom.de](mailto:axel.clauberg@telekom.de)

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

Email: [lhoffman@bouyguestelecom.fr](mailto: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](mailto:fibrib@gmail.com)

#### 14. Acknowledgement

The authors would like to thank Ole Troan, Ralph Droms 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].

## 15. Appendix: Alternatives for Port-Restricted Address Allocation

Besides DHCPv4, other alternatives for address and port-set provisioning, e.g.- PCP, DHCPv6, IPCP, MAY also be implemented.

- o PCP[I-D.ietf-pcp-base]: an initiator MAY use [I-D.tsou-pcp-natcoord] to retrieve a restricted IPv4 address and a set of ports.
- o DHCPv6: the DHCPv6 protocol MAY be extended to support port-set allocation [I-D.boucadair-dhcpv6-shared-address-option], along with IPv6-mapped IPv4 address allocation.
- o IPCP: IPCP MAY be extended to carry the port-set (e.g., [RFC6431]).

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

## 16. References

### 16.1. 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.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [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.

## 16.2. Informative References

- [I-D.bajko-pripaddrassign]  
Bajko, G., Savolainen, T., Boucadair, M., and P. Levis, "Port Restricted IP Address Assignment", draft-bajko-pripaddrassign-04 (work in progress), April 2012.
- [I-D.boucadair-dhcpv6-shared-address-option]  
Boucadair, M., Levis, P., Grimault, J., Savolainen, T., and G. Bajko, "Dynamic Host Configuration Protocol (DHCPv6) Options for Shared IP Addresses Solutions", draft-boucadair-dhcpv6-shared-address-option-01 (work in progress), December 2009.
- [I-D.cui-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-06 (work in progress), May 2012.
- [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-ipv6-03 (work in progress), May 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-26 (work in progress), June 2012.

- [I-D.ietf-softwire-public-4over6]  
Cui, Y., Wu, J., Wu, P., Metz, C., Vautrin, O., and Y. Lee, "Public IPv4 over IPv6 Access Network", draft-ietf-softwire-public-4over6-01 (work in progress), March 2012.
- [I-D.mdt-softwire-mapping-address-and-port]  
Bao, C., Troan, O., Matsushima, S., Murakami, T., and X. Li, "Mapping of Address and Port (MAP)", draft-mdt-softwire-mapping-address-and-port-03 (work in progress), January 2012.
- [I-D.mrugalski-softwire-dhcpv4-over-v6-option]  
Mrugalski, T. and P. Wu, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for DHCPv4 over IPv6 Transport", draft-mrugalski-softwire-dhcpv4-over-v6-option-00 (work in progress), April 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.tsou-pcp-natcoord]  
Sun, Q., Boucadair, M., Deng, X., Zhou, C., and T. Tsou, "Using PCP To Coordinate Between the CGN and Home Gateway Via Port Allocation", draft-tsou-pcp-natcoord-05 (work in progress), March 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  
Philadelphia, Bonn 53227  
Germany

Email: ian.farrer@telekom.de



Network Working Group  
Internet-Draft  
Expires: January 15, 2013

M. Xu  
Y. Cui  
J. Wu  
S. Yang  
Tsinghua University  
C. Metz  
G. Shepherd  
Cisco Systems  
July 14, 2012

Software Mesh Multicast  
draft-ietf-softwire-mesh-multicast-03

Abstract

The Internet needs to support IPv4 and IPv6 packets. Both address families and their attendant protocol suites support multicast of the single-source and any-source varieties. As part of the transition to IPv6, there will be scenarios where a backbone network running one IP address family internally (referred to as internal IP or I-IP) will provide transit services to attached client networks running another IP address family (referred to as external IP or E-IP). It is expected that the I-IP backbone will offer unicast and multicast transit services to the client E-IP networks.

Software Mesh is a solution to E-IP unicast and multicast support across an I-IP backbone. This document describes the mechanisms for supporting Internet-style multicast across a set of E-IP and I-IP networks supporting software mesh.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 15, 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.

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 . . . . .	4
2. Terminology . . . . .	5
3. Scenarios of Interest . . . . .	7
3.1. IPv4-over-IPv6 . . . . .	7
3.2. IPv6-over-IPv4 . . . . .	8
4. IPv4-over-IPv6 Mechanism . . . . .	10
4.1. Mechanism Overview . . . . .	10
4.2. Group Address Mapping . . . . .	10
4.3. Source Address Mapping . . . . .	11
4.4. Routing Mechanism . . . . .	12
5. IPv6-over-IPv4 Mechanism . . . . .	14
5.1. Mechanism Overview . . . . .	14
5.2. Group Address Mapping . . . . .	14
5.3. Source Address Mapping . . . . .	14
5.4. Routing Mechanism . . . . .	15
6. Actions performed by AFBR . . . . .	17
6.1. E-IP (*,G) state maintenance . . . . .	17
6.2. E-IP (S,G) state maintenance . . . . .	17
6.3. I-IP (S',G') state maintenance . . . . .	17
6.4. E-IP (S,G,rpt) state maintenance . . . . .	17
6.5. Inter-AFBR signaling . . . . .	17
6.6. Process and forward multicast data . . . . .	19
6.7. SPT switchover . . . . .	19
7. Other Considerations . . . . .	21
7.1. Other PIM Message Types . . . . .	21
7.2. Selecting a Tunneling Technology . . . . .	21
7.3. TTL . . . . .	21
7.4. Fragmentation . . . . .	21
8. Security Considerations . . . . .	22
9. IANA Considerations . . . . .	23
10. References . . . . .	24
10.1. Normative References . . . . .	24
10.2. Informative References . . . . .	24
Appendix A. Acknowledgements . . . . .	25
Authors' Addresses . . . . .	26

## 1. Introduction

The Internet needs to support IPv4 and IPv6 packets. Both address families and their attendant protocol suites support multicast of the single-source and any-source varieties. As part of the transition to IPv6, there will be scenarios where a backbone network running one IP address family internally (referred to as internal IP or I-IP) will provide transit services to attached client networks running another IP address family (referred to as external IP or E-IP).

The preferred solution is to leverage the multicast functions inherent in the I-IP backbone, to efficiently and scalably forward client E-IP multicast packets inside an I-IP core tree, which roots at one or more ingress AFBR nodes and branches out to one or more egress AFBR leaf nodes.

[6] outlines the requirements for the softwires mesh scenario including the multicast. It is straightforward to envisage that client E-IP multicast sources and receivers will reside in different client E-IP networks connected to an I-IP backbone network. This requires that the client E-IP source-rooted or shared tree should traverse the I-IP backbone network.

One method to accomplish this is to re-use the multicast VPN approach outlined in [10]. MVPN-like schemes can support the softwire mesh scenario and achieve a "many-to-one" mapping between the E-IP client multicast trees and 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 a large amount of (S,G) states. Aggregation at the edge contains the (S,G) states that need to be maintained by the network operator supporting the customer VPNs. The disadvantage of this approach is the possible inefficient bandwidth and resource utilization when multicast packets are delivered to a receiver AFBR with no attached E-IP receivers.

Internet-style multicast is somewhat different in that the trees tend to be relatively sparse and source-rooted. The need for multicast aggregation at the edge (where many customer multicast trees are mapped into a few or one backbone multicast trees) does not exist and to date has not been identified. Thus the need for a basic or closer alignment with E-IP and I-IP multicast procedures emerges.

A framework on how to support such methods is described in [8]. In this document, a more detailed discussion supporting the "one-to-one" mapping schemes for the IPv6 over IPv4 and IPv4 over IPv6 scenarios will be discussed.

## 2. Terminology

An example of a softwire mesh network supporting multicast is illustrated in Figure 1. A multicast source *S* is located in one E-IP client network, while candidate E-IP group receivers are located in the same or different E-IP client networks that all share a common I-IP transit network. When E-IP sources and receivers are not local to each other, they can only communicate with each other through the I-IP core. There may be several E-IP sources for some multicast group residing in different client E-IP networks. In the case of shared trees, the E-IP sources, receivers and RPs might be located in different client E-IP networks. In a simple case the resources of the I-IP core are managed by a single operator although the inter-provider case is not precluded.

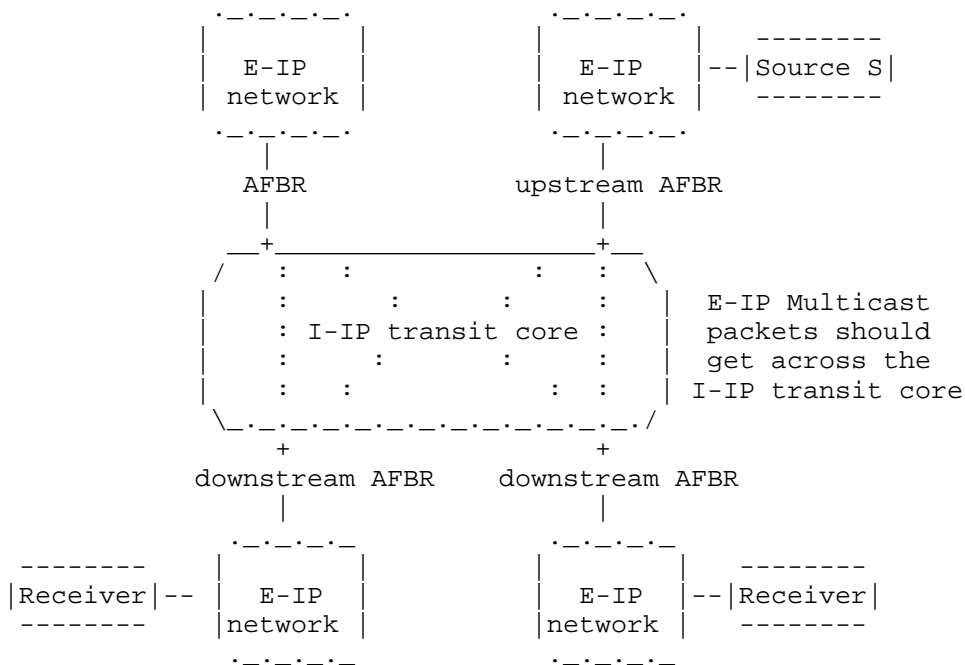


Figure 1: Softwire Mesh Multicast Framework

Terminology used in this document:

- o Address Family Border Router (AFBR) - A dual-stack router interconnecting two or more networks using different IP address families. In the context of softwire mesh multicast, the AFBR runs

E-IP and I-IP control planes to maintain E-IP and I-IP multicast states respectively and performs the appropriate encapsulation/decapsulation of client E-IP multicast packets for transport across the I-IP core. An AFBR will act as a source and/or receiver in an I-IP multicast tree.

- o Upstream AFBR: The AFBR router that is located on the upper reaches of a multicast data flow.

- o Downstream AFBR: The AFBR router that is located on the lower reaches of a multicast data flow.

- o I-IP (Internal IP): This refers to the form of IP (i.e., either IPv4 or IPv6) that is supported by the core (or backbone) network. An I-IPv6 core network runs IPv6 and an I-IPv4 core network runs IPv4.

- o E-IP (External IP): This refers to the form of IP (i.e. either IPv4 or IPv6) that is supported by the client network(s) attached to the I-IP transit core. An E-IPv6 client network runs IPv6 and an E-IPv4 client network runs IPv4.

- o I-IP core tree: A distribution tree rooted at one or more AFBR source nodes and branched out to one or more AFBR leaf nodes. An I-IP core tree is built using standard IP or MPLS multicast signaling protocols operating exclusively inside the I-IP core network. An I-IP core tree is used to 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 IPv4-embedded IPv6 source address.

### 3. Scenarios of Interest

This section describes the two different scenarios where softwires mesh multicast will apply.

#### 3.1. IPv4-over-IPv6

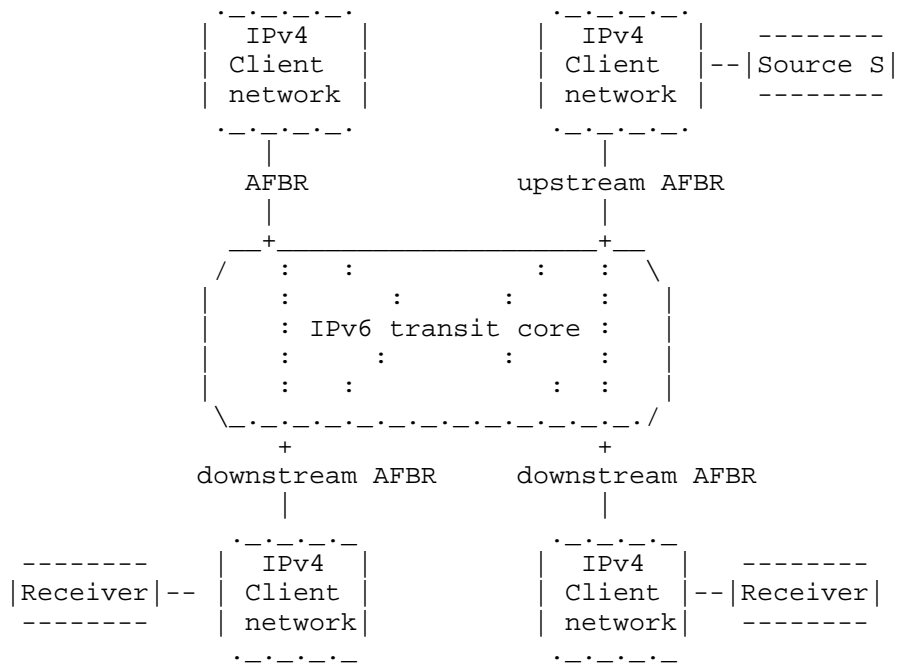


Figure 2: IPv4-over-IPv6 Scenario

In this scenario, the E-IP client networks run IPv4 and I-IP core runs IPv6. This scenario is illustrated in Figure 2.

Because of the much larger IPv6 group address space, it will not be a problem to map individual client E-IPv4 tree to a specific I-IPv6 core tree. This simplifies operations on the AFBR because it becomes possible to algorithmically map an IPv4 group/source address to an IPv6 group/source address and vice-versa.

The IPv4-over-IPv6 scenario is an emerging requirement as network operators build out native IPv6 backbone networks. These networks naturally support native IPv6 services and applications but it is with near 100% certainty that legacy IPv4 networks handling unicast

and multicast should be accommodated.

### 3.2. IPv6-over-IPv4

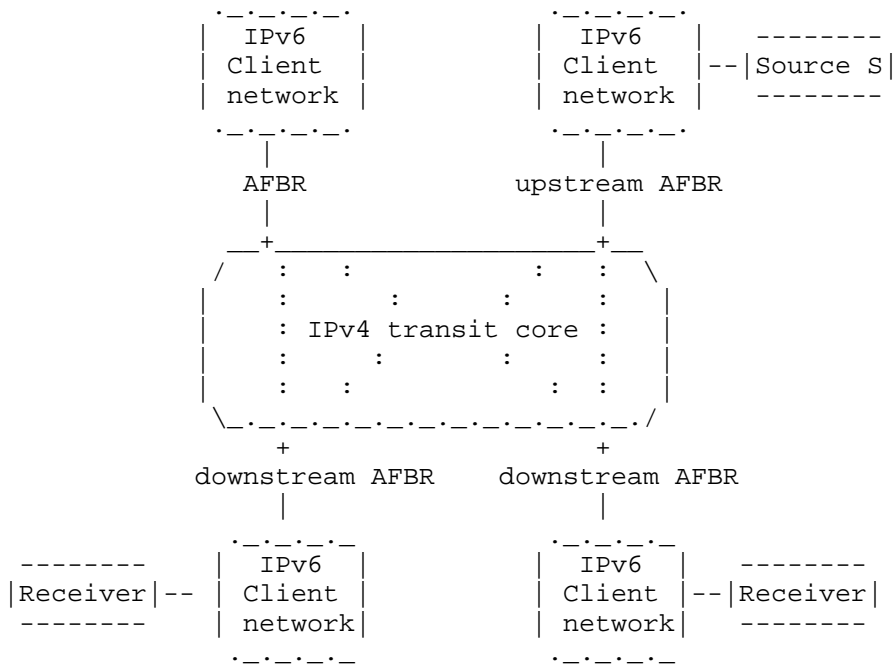


Figure 3: IPv6-over-IPv4 Scenario

In this scenario, the E-IP Client Networks run IPv6 while the I-IP core runs IPv4. This scenario is illustrated in Figure 3.

IPv6 multicast group addresses are longer than IPv4 multicast group addresses. It will not be possible to perform an algorithmic IPv6 - to - IPv4 address mapping without the risk of multiple IPv6 group addresses mapped to the same IPv4 address resulting in unnecessary bandwidth and resource consumption. Therefore additional efforts will be required to ensure that client E-IPv6 multicast packets can be injected into the correct I-IPv4 multicast trees at the AFBRs. This clear mismatch in IPv6 and IPv4 group address lengths means that it will not be possible to perform a one-to-one mapping between IPv6 and IPv4 group addresses unless the IPv6 group address is scoped.

As mentioned earlier, this scenario is common in the MVPN environment. As native IPv6 deployments and multicast applications



emerge from the outer reaches of the greater public IPv4 Internet, it is envisaged that the IPv6 over IPv4 softwire mesh multicast scenario will be a necessary feature supported by network operators.

## 4. IPv4-over-IPv6 Mechanism

### 4.1. Mechanism Overview

Routers in the client E-IPv4 networks contain routes to all other client E-IPv4 networks. Through the set of known and deployed mechanisms, E-IPv4 hosts and routers have discovered or learnt of (S,G) or (\*,G) IPv4 addresses. Any I-IPv6 multicast state instantiated in the core is referred to as (S',G') or (\*,G') and is certainly separated from E-IPv4 multicast state.

Suppose a downstream AFBR receives an E-IPv4 PIM Join/Prune message from the E-IPv4 network for either an (S,G) tree or a (\*,G) tree. The AFBR can translate the E-IPv4 PIM message into an I-IPv6 PIM message with the latter being directed towards I-IP IPv6 address of the upstream AFBR. When the I-IPv6 PIM message arrives at the upstream AFBR, it should be translated back into an E-IPv4 PIM message. The result of these actions is the construction of E-IPv4 trees and a corresponding I-IP tree in the I-IP network.

In this case it is incumbent upon the AFBR routers to perform PIM message conversions in the control plane and IP group address conversions or mappings in the data plane. It becomes possible to devise an algorithmic one-to-one IPv4-to-IPv6 address mapping at AFBRs.

### 4.2. Group Address Mapping

For IPv4-over-IPv6 scenario, a simple algorithmic mapping between IPv4 multicast group addresses and IPv6 group addresses is supported. [11] has already defined an applicable format. Figure 4 is the reminder of the format:

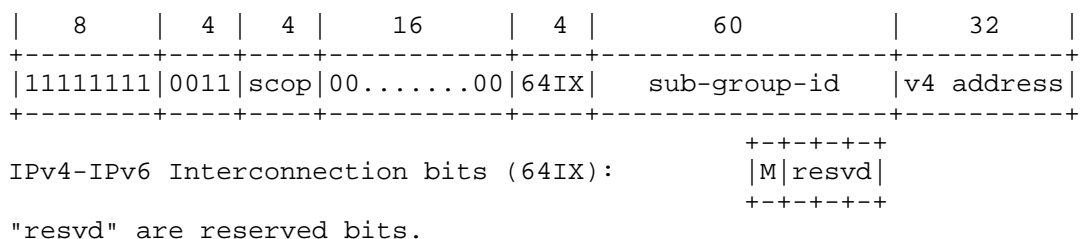


Figure 4: IPv4-Embedded IPv6 Multicast Address Format: SSM Mode

The high order bits of the I-IPv6 address range will be fixed for

mapping purposes. With this scheme, each IPv4 multicast address can be mapped into an IPv6 multicast address (with the assigned prefix), and each IPv6 multicast address with the assigned prefix can be mapped into IPv4 multicast address.

#### 4.3. Source Address Mapping

There are two kinds of multicast --- ASM and SSM. Considering that I-IP network and E-IP network may support different kind of multicast, the source address translation rules could be very complex to support all possible scenarios. But since SSM can be implemented with a strict subset of the PIM-SM protocol mechanisms [5], we can treat I-IP core as SSM-only to make it as simple as possible, then there remains only two scenarios to be discussed in detail:

- o E-IP network supports SSM

One possible way to make sure that the translated I-IPv6 PIM message reaches upstream AFBR is to set S' to a virtual IPv6 address that leads to the upstream AFBR. Figure 5 is the recommended address format based on [9]:

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0-----32--40--48--56--64--72--80--88--96-----127|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   prefix   |v4(32)           | u | suffix   |source address |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|<-----uPrefix64----->|

```

Figure 5: IPv4-Embedded IPv6 Virtual Source Address Format

In this address format, the "prefix" field contains a "Well-Known" prefix or an ISP-defined prefix. An existing "Well-Known" prefix is 64:ff9b, which is defined in [9]; "v4" field is the IP address of one of upstream AFBR's E-IPv4 interfaces; "u" field is defined in [4], and MUST be set to zero; "suffix" field is reserved for future extensions and SHOULD be set to zero; "source address" field stores the original S. We call the overall /96 prefix ("prefix" field and "v4" field and "u" field and "suffix" field altogether) "uPrefix64".

- o E-IP network supports ASM

The (S,G) source list entry and the (\*,G) source list entry only differ in that the latter have both the WC and RPT bits of the Encoded-Source-Address set, while the former all cleared (See Section 4.9.5.1 of [5]). So we can translate source list entries in (\*,G) messages into source list entries in (S',G') messages by applying the format specified in Figure 5 and setting both the WC and RPT bits at upstream AFBRs, and translate them back at upstream AFBRs vice-versa.

#### 4.4. Routing Mechanism

In the mesh multicast scenario, routing information is required to be distributed among AFBRs to make sure that PIM messages that a downstream AFBR propagates reach the right upstream AFBR.

To make it feasible, the /32 prefix in "IPv4-Embedded IPv6 Virtual Source Address Format" must be known to every AFBR, and every AFBR should not only announce the IP address of one of its E-IPv4 interfaces presented in the "v4" field to other AFBRs by MPBGP, but also announce the corresponding uPrefix64 to the I-IPv6 network. Since every IP address of upstream AFBR's E-IPv4 interface is different from each other, every uPrefix64 that AFBR announces should be different either, and uniquely identifies each AFBR. "uPrefix64" is an IPv6 prefix, and the distribution of it is the same as the distribution in the traditional mesh unicast scenario. But since "v4" field is an E-IPv4 address, and BGP messages are NOT tunneled through softwires or through any other mechanism as specified in [8], AFBRs MUST be able to transport and encode/decode BGP messages that are carried over I-IPv6, whose NLRI and NH are of E-IPv4 address family.

In this way, when a downstream AFBR receives an E-IPv4 PIM (S,G) message, it can translate this message into (S',G') by looking up the IP address of the corresponding AFBR's E-IPv4 interface. Since the uPrefix64 of S' is unique, and is known to every router in the I-IPv6 network, the translated message will eventually arrive at the corresponding upstream AFBR, and the upstream AFBR can translate the message back to (S,G). When a downstream AFBR receives an E-IPv4 PIM (\*,G) message, S' can be generated according to the format specified in Figure 4, with "source address" field set to \*(the IPv4 address of RP). The translated message will eventually arrive at 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 (see Section 4.7 of [5]), when this upstream AFBR checks the "source address" field of the message, it'll find the IPv4 address of RP, so this upstream AFBR judges that this is originally a (\*,G) message, then it translates the message back to the (\*,G)

message and processes it.

## 5. IPv6-over-IPv4 Mechanism

### 5.1. Mechanism Overview

Routers in the client E-IPv6 networks contain routes to all other client E-IPv6 networks. Through the set of known and deployed mechanisms, E-IPv6 hosts and routers have discovered or learnt of (S,G) or (\*,G) IPv6 addresses. Any I-IP multicast state instantiated in the core is referred to as (S',G') or (\*,G') and is certainly separated from E-IP multicast state.

This particular scenario introduces unique challenges. Unlike the IPv4-over-IPv6 scenario, it's impossible to map all of the IPv6 multicast address space into the IPv4 address space to address the one-to-one Softwire Multicast requirement. To coordinate with the "IPv4-over-IPv6" scenario and keep the solution as simple as possible, one possible solution to this problem is to limit the scope of the E-IPv6 source addresses for mapping, such as applying a "Well-Known" prefix or an ISP-defined prefix.

### 5.2. Group Address Mapping

To keep one-to-one group address mapping simple, the group address range of E-IP IPv6 can be reduced in a number of ways to limit the scope of addresses that need to be mapped into the I-IP IPv4 space.

A recommended multicast address format is defined in [11]. The high order bits of the E-IPv6 address range will be fixed for mapping purposes. With this scheme, each IPv4 multicast address can be mapped into an IPv6 multicast address (with the assigned prefix), and each IPv6 multicast address with the assigned prefix can be mapped into IPv4 multicast address.

### 5.3. Source Address Mapping

There are two kinds of multicast --- ASM and SSM. Considering that I-IP network and E-IP network may support different kind of multicast, the source address translation rules could be very complex to support all possible scenarios. But since SSM can be implemented with a strict subset of the PIM-SM protocol mechanisms [5], we can treat I-IP core as SSM-only to make it as simple as possible, then there remains only two scenarios to be discussed in detail:

- o E-IP network supports SSM

To make sure that the translated I-IPv4 PIM message reaches the upstream AFBR, we need to set S' to an IPv4 address that leads to the upstream AFBR. But due to the non-"one-to-one" mapping of

E-IPv6 to I-IPv4 unicast address, the upstream AFBR is unable to remap the I-IPv4 source address to the original E-IPv6 source address without any constraints.

We apply a fixed IPv6 prefix and static mapping to solve this problem. A recommended source address format is defined in [9]. Figure 6 is the reminder of the format:

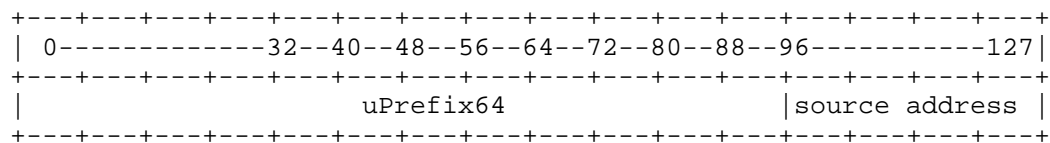


Figure 6: IPv4-Embedded IPv6 Source Address Format

In this address format, the "uPrefix64" field starts with a "Well-Known" prefix or an ISP-defined prefix. An existing "Well-Known" prefix is 64:ff9b/32, which is defined in [9]; "source address" field is the corresponding I-IPv4 source address.

- o E-IP network supports ASM

The (S,G) source list entry and the (\*,G) source list entry only differ in that the latter have both the WC and RPT bits of the Encoded-Source-Address set, while the former all cleared (See Section 4.9.5.1 of [5]). So we can translate source list entries in (\*,G) messages into source list entries in (S'G') messages by applying the format specified in Figure 5 and setting both the WC and RPT bits at upstream AFBRs, and translate them back at upstream AFBRs vice-versa. Here, the E-IPv6 address of RP MUST follow the format specified in Figure 6. RP' is the upstream AFBR that locates between RP and the downstream AFBR.

#### 5.4. Routing Mechanism

In the mesh multicast scenario, routing information is required to be distributed among AFBRs to make sure that PIM messages that a downstream AFBR propagates reach the right upstream AFBR.

To make it feasible, the /96 uPrefix64 must be known to every AFBR, every E-IPv6 address of sources that support mesh multicast MUST follow the format specified in Figure 6, and the corresponding

upstream AFBR of this source should announce the I-IPv4 address in "source address" field of this source's IPv6 address to the I-IPv4 network. Since uPrefix64 is static and unique in IPv6-over-IPv4 scenario, there is no need to distribute it using BGP. The distribution of "source address" field of multicast source addresses is a pure I-IPv4 process and no more specification is needed.

In this way, when a downstream AFBR receives a (S,G) message, it can translate the message into (S',G') by simply taking off the prefix in S. Since S' is known to every router in I-IPv4 network, the translated message will eventually arrive at the corresponding upstream AFBR, and the upstream AFBR can translate the message back to (S,G) by appending the prefix to S'. When a downstream AFBR receives a (\*,G) message, it can translate it into (S',G') by simply taking off the prefix in \*(the E-IPv6 address of RP). Since S' is known to every router in I-IPv4 network, the translated message will eventually arrive at RP'. And since every PIM router within a PIM domain must be able to map a particular multicast group address to the same RP (see Section 4.7 of [5]), RP' knows that S' is the mapped I-IPv4 address of RP, so RP' will translate the message back to (\*,G) by appending the prefix to S' and propagate it towards RP.



## 6. Actions performed by AFBR

The following actions are performed by AFBRs:

### 6.1. E-IP (\*,G) state maintenance

When an AFBR wishes to propagate a Join/Prune(\*,G) message to an I-IP upstream router, the AFBR MUST translate Join/Prune(\*,G) messages into Join/Prune(S',G') messages following the rules specified above, then send the latter.

### 6.2. E-IP (S,G) state maintenance

When an AFBR wishes to propagate a Join/Prune(S,G) message to an I-IP upstream router, the AFBR MUST translate Join/Prune(S,G) messages into Join/Prune(S',G') messages following the rules specified above, then send the latter.

### 6.3. I-IP (S',G') state maintenance

It is possible that there runs a non-transit I-IP PIM-SSM in the I-IP transit core. Since the translated source address starts with the unique "Well-Known" prefix or the ISP-defined prefix that should not be used otherwise, mesh multicast won't influence non-transit PIM-SM multicast at all. When one AFBR receives an I-IP (S',G') message, it should check S'. If S' starts with the unique prefix, it means that this message is actually a translated E-IP (S,G) or (\*,G) message, then the AFBR should translate this message back to E-IP PIM message and process it.

### 6.4. E-IP (S,G,rpt) state maintenance

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

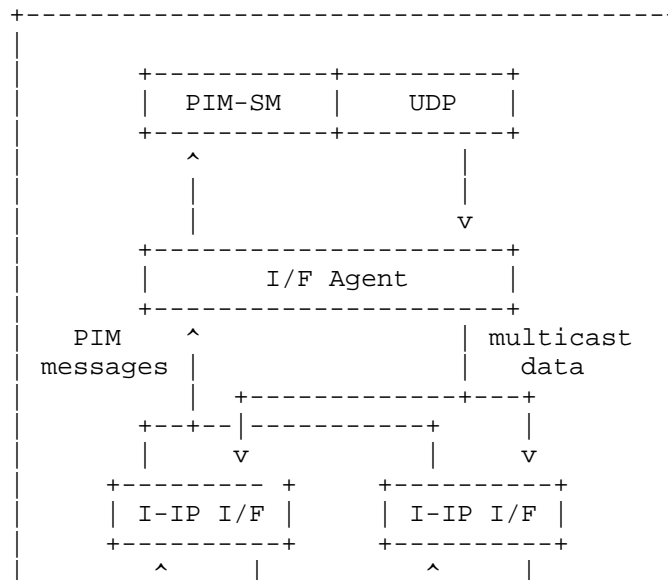
### 6.5. Inter-AFBR signaling

Assume that one downstream AFBR has joined a RPT of (\*,G) and a SPT of (S,G), and decide to perform a SPT switchover. According to [5], it should propagate a Prune(S,G,rpt) message along with the periodical Join(\*,G) message upstream towards RP. Unfortunately, routers in I-IP transit core are not supposed to understand (S,G,rpt) messages since I-IP transit core is treated as SSM-only. As a result, this downstream AFBR is unable to prune S from this RPT, then it will receive two copies of the same data of (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 RPT.

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

When RP' receives this encapsulated message, it should decapsulate this message as what it does in the unicast scenario, and get the original (S,G,rpt) message. The incoming interface of this message may be different from 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 [5] on the incoming interface.

To solve this problem, and keep the solution as simple as possible, we introduce an "interface agent" to process all the encapsulated (S,G,rpt) messages the upstream AFBR receives, and prune S from the RPT of group G when no downstream AFBR wants to receive multicast data of (S,G) along the RPT. In this way, we do insure that downstream AFBRs won't miss any multicast data that they needs, at the cost of duplicated multicast data of (S,G) along the RPT received by SPT-switched-over downstream AFBRs, if there exists at least one downstream AFBR that hasn't yet sent Prune(S,G,rpt) messages to the upstream AFBR. The following diagram shows an example of how an "interface agent" may be implemented:



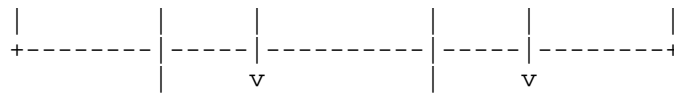


Figure 7: Interface Agent Implementation Example

In this example, the interface agent has two responsibilities: In the control plane, it should work as a real interface that has joined  $(*,G)$  in representative of all the I-IP interfaces who should have been outgoing interfaces of  $(*,G)$  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 of every downstream AFBR, and submits  $Prune(S,G,rpt)$  messages to the PIM-SM module only when every  $(S,G,rpt)$  state machine is at  $Prune(P)$  or  $PruneTmp(P')$  state, which means that no downstream AFBR wants to receive multicast data of  $(S,G)$  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 changed its mind to receive multicast data of  $(S,G)$  along the RPT again, the interface agent should send a  $Join(S,G,rpt)$  to PIM-SM module immediately; In the data plane, upon receiving a multicast data packet, the interface agent should encapsulate it at first, then propagate the encapsulated packet onto every I-IP interface.

NOTICE: There may exist an E-IP neighbor of  $RP'$  that has joined the RPT of  $G$ , so the per-interface state machine for receiving E-IP  $Join/Prune(S,G,rpt)$  messages should still take effect.

#### 6.6. Process and forward multicast data

On receiving multicast data from upstream routers, the AFBR looks up its forwarding table to check the IP address of each outgoing interface. If there exists at least one outgoing interface whose IP address family is different from the incoming interface, the AFBR should encapsulate/decapsulate this packet and forward it to such outgoing interface(s), then forward the data to other outgoing interfaces without encapsulation/decapsulation.

When a downstream AFBR that has already switched over to SPT of  $S$  receives an encapsulated multicast data packet of  $(S,G)$  along the RPT, it should silently drop this packet.

#### 6.7. SPT switchover

After a new AFBR expresses its interest in receiving 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 SPT of some source(s) or not.

To minimize this redundancy, it's recommended that every AFBR's SwitchToSptDesired(S,G) function employs the "switch on first packet" policy. In this way, the delay of switchover to SPT is kept as little 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 redundancy will be produced.

## 7. Other Considerations

### 7.1. Other PIM Message Types

Apart from Join or Prune, there exists other message types 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 routers on a link to negotiate with each other. They don't need to be translated for forwarding, thus the process of these messages is out of scope for this document.

### 7.2. Selecting a Tunneling Technology

The choice of tunneling technology is a matter of policy configured at AFBRs. It's recommended that all AFBRs use the same technology, otherwise some AFBRs may not be able to decapsulate encapsulated packets from other AFBRs that use a different tunneling technology.

### 7.3. TTL

The process of TTL depends on the tunneling technology, and is out of scope for this document.

### 7.4. Fragmentation

The encapsulation performed by upstream AFBR will increase the size of packets. As a result, the outgoing I-IP link MTU may not accommodate the extra size. As it's not always possible for core operators to increase every link's MTU, fragmentation and reassembling of encapsulated packets MUST be supported by AFBRs.

## 8. Security Considerations

The AFBR routers could maintain secure communications through the use of Security Architecture for the Internet Protocol as described in [RFC4301]. But when adopting some schemes that will cause heavy burden on routers, some attacker may use it as a tool for DDoS attack.

## 9. IANA Considerations

When AFBRs perform address mapping, they should follow some predefined rules, especially the IPv6 prefix for source address mapping should be predefined, so that ingress AFBR and egress AFBR can finish the mapping procedure correctly. The IPv6 prefix for translation can be unified within only the transit core, or within global area. In the later condition, the prefix should be assigned by IANA.

## 10. References

### 10.1. Normative References

- [1] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.
- [2] Foster, B. and F. Andreassen, "Media Gateway Control Protocol (MGCP) Redirect and Reset Package", RFC 3991, February 2005.
- [3] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.
- [4] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [5] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.
- [6] Li, X., Dawkins, S., Ward, D., and A. Durand, "Softwire Problem Statement", RFC 4925, July 2007.
- [7] Wijnands, IJ., Boers, A., and E. Rosen, "The Reverse Path Forwarding (RPF) Vector TLV", RFC 5496, March 2009.
- [8] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, June 2009.
- [9] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [10] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", RFC 6513, February 2012.

### 10.2. Informative References

- [11] Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X., and M. Xu, "IPv6 Multicast Address Format With Embedded IPv4 Multicast Address", draft-ietf-mboned-64-multicast-address-format-02 (work in progress), May 2012.



#### Appendix A. Acknowledgements

Wenlong Chen, Xuan Chen, Alain Durand, Yiu Lee, Jacni Qin and Stig Venaas provided useful input into this document.

## Authors' Addresses

Mingwei Xu  
Tsinghua University  
Department of Computer Science, Tsinghua University  
Beijing 100084  
P.R. China

Phone: +86-10-6278-5822  
Email: xmw@cernet.edu.cn

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

Phone: +86-10-6278-5822  
Email: cuiyong@tsinghua.edu.cn

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

Phone: +86-10-6278-5822  
Email: yangshu@csnet1.cs.tsinghua.edu.cn

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

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

Greg Shepherd  
Cisco Systems  
170 West Tasman Drive  
San Jose, CA 95134  
USA

Phone: +1-541-912-9758  
Email: shep@cisco.com



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: January 10, 2013

S. Tsuchiya, Ed.  
Cisco Systems  
S. Ohkubo  
Sakura Internet  
Y. Kawakami  
INTERNET MULTIFEED CO.  
July 9, 2012

Stateless IPv4 over IPv6 report  
draft-janog-softwire-report-00

Abstract

Stateless IPv4 over IPv6 tunnel such as MAP(Mapping of Address and Port)/4rd(IPv4 Residual Deployment) designs to support IPv4 over IPv6 island and resolve IPv4 shortage problem by Address and Port Mapping technique. This document describes supported vendor's implementation, ipv4 functionality over IPv6 and interoperability report.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 10, 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 . . . . .	4
2. Implementation Report . . . . .	4
2.1. IpInfusion [IPI] . . . . .	4
2.2. FURUKAWA NETWORK SOLUTION [FNSC] . . . . .	4
2.3. Internet Initiative Japan [SEIL] . . . . .	4
3. IPv4 functionality over IPv6 . . . . .	5
3.1. T-1:ICMP . . . . .	5
3.2. T-2:IPSec VPN . . . . .	6
3.3. T-3:SSL VPN . . . . .	6
3.4. T-4:FTP . . . . .	7
3.5. T-5:PPTP . . . . .	7
3.6. T-6:L2TP . . . . .	7
3.7. T-7:Instant Messaging and VoIP . . . . .	8
3.7.1. Facebook on the web (http) . . . . .	8
3.7.2. Facebook via a client (xmpp) . . . . .	8
3.7.3. Jabber.org chat service (xmpp) . . . . .	8
3.7.4. Gmail chat on the web (http) . . . . .	8
3.7.5. Gmail chat via a client (xmpp) . . . . .	9
3.7.6. Google Talk client . . . . .	9
3.7.7. AIM (AOL) . . . . .	9
3.7.8. ICQ (AOL) . . . . .	9
3.7.9. Skype . . . . .	10
3.7.10. MSN . . . . .	10
3.7.11. Webex . . . . .	10
3.7.12. Sametime . . . . .	10
3.7.13. facetime . . . . .	11
3.8. T-8:NAT veification tool . . . . .	11
3.9. Test Result Summary . . . . .	12
4. Interoperability . . . . .	12
5. Next step . . . . .	12
6. Contributors . . . . .	12
7. Acknowledgements . . . . .	13
8. IANA Considerations . . . . .	13
9. Security Considerations . . . . .	13
10. References . . . . .	13
10.1. Normative References . . . . .	13
10.2. Informative References . . . . .	14
Appendix A. Additional Stuff . . . . .	16
A.1. test network topology and parameters . . . . .	16
A.2. Configuration . . . . .	16
A.2.1. IPI . . . . .	16
A.2.2. FNSC:BR . . . . .	17
A.2.3. FNSC:CE . . . . .	18
A.2.4. SEIL . . . . .	19
Authors' Addresses . . . . .	20

## 1. Introduction

Stateless IPv4 over IPv6 tunnel such as MAP(Mapping of Address and Port)/4rd(IPv4 Residual Deployment) designs to support IPv4 over IPv6 island and resolve IPv4 shortage problem by Address and Port Mapping technique. Japan Network Operators Group[JANOG] has presented 4rd[draft-murakami-softwire-4rd] test environment to the JANOG30 conference network. This document describes 4rd[draft-murakami-softwire-4rd] supported vendor's implementation, ipv4 functionality over IPv6 and interoperability report.

## 2. Implementation Report

Three vendors participated to the JANOG30 4rd[draft-murakami-softwire-4rd] test network. In this session, describes information of "implementation reference" and "provisioning method".

### 2.1. IpInfusion [IPI]

- implementation reference

- [draft-murakami-softwire-4rd-00]

- provisioning method

- CLI configuration

### 2.2. FURUKAWA NETWORK SOLUTION [FNCS]

- implementation reference

- [draft-murakami-softwire-4rd-00]

- provisioning method

- CLI configuration

### 2.3. Internet Initiative Japan [SEIL]

- implementation reference

- [draft-murakami-softwire-4rd-00]

- provisioning method

- CLI configuration



### 3. IPv4 functionality over IPv6

4rd [draft-murakami-softwire-4rd] uses A+P technologies with NAT44. Basic NAT requirement is defined as [RFC4787], [RFC5508] and [RFC5382]. But there is difference of implementation among vendors. ALG support also depends on vendors implementations. This test result is feedback from operators community [JANOG] to vendors and IETF community.

#### 3.1. T-1:ICMP

Section 9 of [draft-murakami-softwire-4rd] describes about ICMP handling in 4rd domain.

IPv4 MTU configured 1460 in tunnel interface, because the conference network was well-managed. It describes on Section 6 of [draft-murakami-softwire-4rd]

T-1-1: echo/echo reply

Test result of ICMP echo/echo reply between 4rd[draft-murakami-softwire-4rd] network and global internet.

+-----+	+-----+	+-----+	+-----+
Vendor	[IPI]	[FNSC]	[SEIL]
+-----+	+-----+	+-----+	+-----+
Result	OK	OK	OK
+-----+	+-----+	+-----+	+-----+

T-1-2: does not response

Test result of ICMP host unreachable message from global internet to 4rd[draft-murakami-softwire-4rd].

+-----+	+-----+	+-----+	+-----+
Vendor	[IPI]	[FNSC]	[SEIL]
+-----+	+-----+	+-----+	+-----+
Result	OK	OK	OK
+-----+	+-----+	+-----+	+-----+

T-1-3: ttl expire

Test result of ICMP ttl expire message from global internet to 4rd[draft-murakami-softwire-4rd].

Vendor	[IPI]	[FNSC]	[SEIL]
Result	OK	OK	OK

## T-1-4: Packet too Big

Could not confirm Packet too big message from 4rd[draft-murakami-softwire-4rd] CE, because test terminal MTU size was 1300 even though 4rd[draft-murakami-softwire-4rd] CE's IPv4 MTU was configured as 1460.

Vendor	[IPI]	[FNSC]	[SEIL]
Result	N/A	N/A	N/A

## 3.2. T-2:IPSec VPN

IPSec VPN [RFC2401] uses ESP packets, therefore the client should support NAT traversal [RFC3948] under 4rd enviroment.

## T-2-1:IPSec

Vendor	[IPI]	[FNSC]	[SEIL]
Result	N.G	N.G	N.G

This result is expected behavior.

## T-2-2:IPSec VPN(UDP:NAT Traversal)

Vendor	[IPI]	[FNSC]	[SEIL]
Result	O.K	O.K	O.K

## 3.3. T-3:SSL VPN

It should be no problem, because SSL VPN[RFC4347] uses TCP sockets.

Vendor	[IPI]	[FNSC]	[SEIL]
Result	O.K	O.K	O.K

### 3.4. T-4:FTP

FTP[RFC0959] PORT(Active) and PASV(Passive) mode had sometimes problem in NAT44. [RFC2428] is enhancement FTP for IPv6/NAT. But 4rd [draft-murakami-softwire-4rd] devices need to support FTP ALG, if the implementation would not be deployed in widely.

T-4-1:Passive(PASV) mode

Vendor	[IPI]	[FNSC]	[SEIL]
Result	O.K	O.K	O.K

T-4-2:Active(PORT) mode

Vendor	[IPI]	[FNSC]	[SEIL]
Result	N.G	O.K	O.K

[IPI] does not support FTP Active mode in this configuration.

### 3.5. T-5:PPTP

PPTP[RFC2637] uses GRE and TCP port 1723. Unless configuring to pass GRE and TCP port 1723, can not use PPTP on 4rd [draft-murakami-softwire-4rd].

Vendor	[IPI]	[FNSC]	[SEIL]
Result	N.G	N.G	N.G

### 3.6. T-6:L2TP

L2TP/IPsec[RFC3193] should support on 4rd[draft-murakami-softwire-4rd] using with NAT Traversal[RFC3948].

Vendor	[IPI]	[FNSC]	[SEIL]
Result	O.K	O.K	O.K

### 3.7. T-7:Instant Messaging and VoIP

Verified functionality of Instant Messaging and VoIP tool that described on section 5.3 of [RFC6586] and facetime within same 4rd CE, different 4rd CEs and between 4rd BR and CE.

#### 3.7.1. Facebook on the web (http)

Combination	[IPI]	[FNSC]	[SEIL]	Internet
[IPI]	O.K	O.K	O.K	O.K
[FNSC]	O.K	O.K	O.K	O.K
[SEIL]	O.K	O.K	O.K	O.K

Tested both chat and video.

#### 3.7.2. Facebook via a client (xmpp)

Combination	[IPI]	[FNSC]	[SEIL]	Internet
[IPI]	O.K	O.K	O.K	O.K
[FNSC]	O.K	O.K	O.K	O.K
[SEIL]	O.K	O.K	O.K	O.K

#### 3.7.3. Jabber.org chat service (xmpp)

Not tested

#### 3.7.4. Gmail chat on the web (http)

Combination	[IPI]	[FNSC]	[SEIL]	Internet
[IPI]	O.K	O.K	O.K	O.K
[FNSC]	O.K	O.K	O.K	O.K
[SEIL]	O.K	O.K	O.K	O.K

Tested chat,voice and video.

#### 3.7.5. Gmail chat via a client (xmpp)

Combination	[IPI]	[FNSC]	[SEIL]	Internet
[IPI]	O.K	O.K	O.K	O.K
[FNSC]	O.K	O.K	O.K	O.K
[SEIL]	O.K	O.K	O.K	O.K

#### 3.7.6. Google Talk client

Combination	[IPI]	[FNSC]	[SEIL]	Internet
[IPI]	O.K	O.K	O.K	O.K
[FNSC]	O.K	O.K	O.K	O.K
[SEIL]	O.K	O.K	O.K	O.K

Tested chat and voice.

#### 3.7.7. AIM (AOL)

Combination	[IPI]	[FNSC]	[SEIL]	Internet
[IPI]	O.K	O.K	O.K	O.K
[FNSC]	O.K	O.K	O.K	O.K
[SEIL]	O.K	O.K	O.K	O.K

Tested chat and video.

#### 3.7.8. ICQ (AOL)

Combination	[IPI]	[FNSC]	[SEIL]	Internet
[IPI]	O.K	O.K	O.K	O.K
[FNSC]	O.K	O.K	O.K	O.K
[SEIL]	O.K	O.K	O.K	O.K

Tested chat,voice and video

## 3.7.9. Skype

Combination	[IPI]	[FNSC]	[SEIL]	Internet
[IPI]	O.K	O.K	O.K	O.K
[FNSC]	O.K	O.K	O.K	O.K
[SEIL]	O.K	O.K	O.K	O.K

Tested chat,voice and video.

## 3.7.10. MSN

Combination	[IPI]	[FNSC]	[SEIL]	Internet
[IPI]	O.K	O.K	O.K	O.K
[FNSC]	O.K	O.K	O.K	O.K
[SEIL]	O.K	O.K	O.K	O.K

Tested chat,voice and video.

## 3.7.11. Webex

Combination	[IPI]	[FNSC]	[SEIL]	Internet
[IPI]	O.K	O.K	O.K	O.K
[FNSC]	O.K	O.K	O.K	O.K
[SEIL]	O.K	O.K	O.K	O.K

Tested chat,voice and video in the meeting.

## 3.7.12. Sametime

Not tested

## 3.7.13. facetime

Combination	[IPI]	[FNCS]	[SEIL]	Internet
[IPI]	O.K	O.K	O.K	O.K
[FNCS]	O.K	O.K	O.K	O.K
[SEIL]	O.K	O.K	O.K	O.K

## 3.8. T-8:NAT veification tool

Acording to Section-8 of [draft-murakami-softwire-4rd], 4rd CE should support [RFC4787], [RFC5508] and [RFC5382] . This section describes the result of 4rd CEs which were verified by test tool.

## T-8-1:STUN

STUN server and UDP hole punching are used for online game.  
[RFC4787] is Best Current Practise of NAT behavior requirment for UDP.  
[STUN tools] and [public stun server] are useful to confirm  
functionality of online game.

Test item/vendor	[IPI]	[FNCS]	[SEIL]
REQ-1:	Dependent	Independent	Independent
REQ-8:	Not tested	Port Dependent	Port Dependent
REQ-9:	no hairpin	will hairpin	no hairpin

REQ-1: Endpoint-Independent Mapping

REQ-8: Filtering Behavior

REQ-9: Hairpinning

## T-6-2:NAT-Analyzer

[NAT-Analyzer] is JAVA applet in the browser to verify NAT functionality.

[IPI] Result [1]

[FNCS] Result [2]

[SEIL] Result [3]

### 3.9. Test Result Summary

Most of modern applications and VPN protocols could use in multi vendor 4rd [draft-murakami-softwire-4rd]. But there is difference of [RFC4787] test result. Some vendors might be fault when using online games.

### 4. Interoperability

4rd stateless technology that means does not need maintenance of state machine. So there are no problem of interoperability.

But configured ::1 as index of CE IPv6 address even though use shared IPv4 address. A reason is for avoiding the trouble by the difference in the maturity of the anycast support between vendors.

### 5. Next step

We have test plan about MAP[draft-ietf-softwire-map]. The draft will update to result of MAP[draft-ietf-softwire-map] or/and Unified 4rd[draft-ietf-softwire-4rd].

### 6. Contributors

Test network Contributors

Chisato Kashiwagi Chisato.Kashiwagi@ipinfusion.com

Hideaki Hayashi hide@fnsc.co.jp

Hiromitsu Yunoki yunoki@fnsc.co.jp

Kaihei Koyama koyama@kct.co.jp

Kouki Ooyatsu kouki-o@iij.ad.jp

Kunihiro Ishiguro kunihiro.ishiguro@access-company.com

Shuuichi Saito shuu@fnsc.co.jp

Takamasa Ogawa ogawa@kct.co.jp

Takuya Iimura tiimura@cisco.com

Tetsuya Murakami Tetsuya.Murakami@ipinfusion.com



Tomoki Murai murai@fnsc.co.jp

Tomoyuki Sahara tsahara@iiij.ad.jp

Tomoyuki Fukunaga fukunaga@fnsc.co.jp

Naoya Takeda ntakeda@cisco.com

Ryo Sato sr.10005@konami.com

## 7. Acknowledgements

The author would like to thanks JANOG30 participants. The authors would like to thank you Satoru Matsushima, Seiichi Kawamura for their thorough review and comments.

## 8. IANA Considerations

This document has no actions for IANA.

## 9. Security Considerations

There is no additional security requirement.

## 10. References

### 10.1. Normative References

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

Despres, R., Penno, R., Lee, Y., Chen, G., and S. Jiang, "IPv4 Residual Deployment via IPv6 - a unified Stateless Solution (4rd)", draft-ietf-softwire-4rd-00 (work in progress), May 2012.

[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-00 (work in progress), June 2012.

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

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

- [RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [RFC2637] Hamzeh, K., Pall, G., Verthein, W., Taarud, J., Little, W., and G. Zorn, "Point-to-Point Tunneling Protocol", RFC 2637, July 1999.
- [RFC3193] Patel, B., Aboba, B., Dixon, W., Zorn, G., and S. Booth, "Securing L2TP using IPsec", RFC 3193, November 2001.
- [RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.
- [RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.
- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.
- [RFC5508] Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT Behavioral Requirements for ICMP", BCP 148, RFC 5508, April 2009.
- [RFC6586] Arkko, J. and A. Keranen, "Experiences from an IPv6-Only Network", RFC 6586, April 2012.

## 10.2. Informative References

- [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.
- [JANOG] "Japan Network Operators Group",  
<<http://www.janog.gr.jp/en>>.
- [NAT-Analyzer]  
"Network Measurement Activities at TUM",

<<http://natatest.net.in.tum.de/>>.

- [RFC3849] Huston, G., Lord, A., and P. Smith, "IPv6 Address Prefix Reserved for Documentation", RFC 3849, July 2004.
- [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.
- [RFC5387] Touch, J., Black, D., and Y. Wang, "Problem and Applicability Statement for Better-Than-Nothing Security (BTNS)", RFC 5387, November 2008.
- [RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, January 2010.
- [RFC5737] Arkko, J., Cotton, M., and L. Vegoda, "IPv4 Address Blocks Reserved for Documentation", RFC 5737, January 2010.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [RFC6104] Chown, T. and S. Venaas, "Rogue IPv6 Router Advertisement Problem Statement", RFC 6104, February 2011.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.
- [STUN tools]  
    "STUN Client and Server",  
    <<http://sourceforge.net/projects/stun/>>.
- [public stun server]  
    "stunserver.org", <<http://stunserver.org>>.

#### URIs

- [1] <<http://natatest.net.in.tum.de/individualResult.php?hash=c8797328af6487a45bfc6d518e19b605>>
- [2] <<http://natatest.net.in.tum.de/>>

individualResult.php?hash=332166a0250ffa522ef776d837a62221>

[3] <<http://natatest.net.in.tum.de/individualResult.php?hash=d85f28343643bd82f9e1413ca7043564>>

## Appendix A. Additional Stuff

### A.1. test network topology and parameters

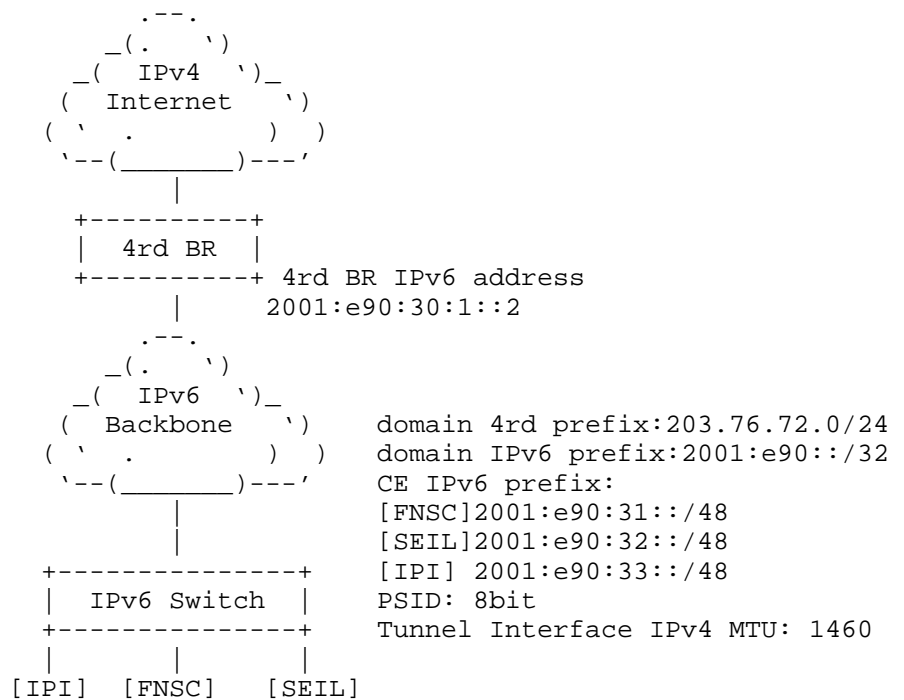


Figure 1

### A.2. Configuration

#### A.2.1. IPI

```
# /etc/rc.local
ip -f inet6 tunnel add map1
ip -f inet6 tunnel change map1 map_mode 4rd
wan_if_name lo
map_border_router 2001:e90:30:1::2
rule_ipv6_suffix 1
rule_ipv4_prefix 203.76.72.0/24
allow_private 1
rule_ipv6_prefix 2001:e90::/32
rule_eabits_length 16
rule_ipv6_iid ::1
map_autosetgw 1
map_autosetaddr 1
mss auto
# /etc/sysctl.conf
# 4RD Configuration
net.ipv4.conf.all.forwarding=1
net.ipv6.conf.all.nd_proxy_loopback=1
net.xrd.delegated_prefix=2001:e90:33::/48
```

#### A.2.2. FNSC:BR

```
ip route 0.0.0.0 0.0.0.0 211.125.127.97
ip route 203.76.72.0 255.255.255.0 tunnel 1
!
ipv6 route ::/0 2001:e90:30:1::1
!
interface GigaEthernet 1/9
 channel-group 19
exit
!
!
interface port-channel 19
 ip address 211.125.127.98 255.255.255.252
 ipv6 address 2001:e90:30:1::2/64
exit
!
interface tunnel 1
 tunnel mode ipinip ipv4 ipv6-tunnel-profile 1
exit
!
ipv4 ipv6-tunnel-profile 1
 profile-mode 4rd
 ipv4-global-prefix 203.76.72.0/24
 ipv6-global-prefix 2001:e90::/32
 user-len 16
 source-address 2001:e90:30:1::2
 ipv6-local-site-id 1
 ipv6-host-id ::1
exit
!
end
```

## A.2.3. FNSC:CE

```
ip route 0.0.0.0 0.0.0.0 tunnel 1
!
access-list 1 permit any
access-list 10 permit 192.168.1.0 0.0.0.255
!
ipv6 route ::/0 2001:e90:30:1000::1
!
ip nat ap_pool POOL1
  ipv6-tunnel-profile 1
exit
ip ipv6-tunnel-profile 1
  ipv6-global-prefix 2001:e90:31:1::1
  ipv6-prefix-len 32
  ipv4-global-prefix 203.76.72.0 255.255.255.0
  user-len 16
  ipv6-host-id ::1
  ipv6-local-site-id ::1
  br-address 2001:e90:30:1::2
exit
!
interface ewan 1
  ip mtu 1500
  ip address 211.125.127.70 255.255.255.224
  ip nat inside source list 1 interface
  ipv6 address 2001:e90:30:1000::6/64
  ipv6 mtu 1500
exit
interface lan 1
  ip address 192.168.1.1 255.255.255.0
  ip mtu 1500
  ipv6 address address-pool pool1 prefix-length 64 interface-id ::1 local-site-id
  ::1 local-site-id-len 16
  ipv6 mtu 1500
exit
interface tunnel 1
  tunnel mode ipip ipv6-tunnel-profile 1
  tunnel source 2001:e90:31:1::1
  ip mtu 1460
  ip nat inside source list 10 ap_pool POOL1
exit
!
```

#### A.2.4. SEIL

```
interface lan0 add 192.168.2.1/24
interface lan0 add 2001:e90:32:1::1/64
interface lan1 add 2001:e90:30:1000::7/64
interface frd0 mtu 1460
interface frd0 tcp-mss 1420
route add default frd0
route6 add default 2001:e90:30:1000::1
nat napt add private 192.168.2.0-192.168.2.255 interface frd0
nat proxy sip add port 5060 protocol tcpudp
frd mode ce
frd ce-address 2001:e90:32:1::1
frd br-address 2001:e90:30:1::2
frd rule add JANOG external-ipv4-prefix 203.76.72.0/24 internal-ipv6-prefix 2001:
e90::/32 index-length 16 host-id ::1
```

#### Authors' Addresses

Shishio Tsuchiya (editor)  
Cisco Systems  
Midtown Tower, 9-7-1, Akasaka  
Minato-Ku, Tokyo 107-6227  
Japan

Phone: +81 3 6434 6543  
Email: [shtsuchi@cisco.com](mailto:shtsuchi@cisco.com)

Shuichi Ohkubo  
Sakura Internet  
33F Sumitomo fudosan Nishi shinjuku Bldg., 7-20-1 Nishi shinjuku  
Shinjuku-Ku, Tokyo 160-0023  
Japan

Phone: +81 3 5332 7070  
Email: [ohkubo@sakura.ad.jp](mailto:ohkubo@sakura.ad.jp)

Yuya Kawakami  
INTERNET MULTIFEED CO.  
OTEMACHI 1st.SQUARE EAST TOWER, 3F 1-5-1, Otemachi,  
Chiyoda-ku, Tokyo 100-0004  
Japan

Phone: +81 3 3282 1040  
Email: [kawakami@mfeed.ad.jp](mailto:kawakami@mfeed.ad.jp)





Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: January 11, 2013

S. Tsuchiya, Ed.  
J. Qin  
Cisco Systems  
July 10, 2012

IP TUNNEL MIB Extention for software  
draft-shishio-software-rfc4087update-00

## Abstract

This memo defines a Management Information Base (MIB) module for use with network management protocols in the Internet community. In particular, it describes managed objects used for managing tunnels of any type over IPv4 and IPv6 networks.

IP TUNNEL MIB[RFC4087] provides provisioning capability for IPv4 and IPv6 tunnel by SNMP. But it is not enough to support modern tunnel protocol such as 6rd[RFC5969] and MAP[draft-ietf-software-map]. The document describes extension of IP TUNNEL MIB[RFC4087] to support 6rd[RFC5969] and MAP[draft-ietf-software-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 11, 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. The Internet-Standard Management Framework . . . . .	3
3. Conventions . . . . .	3
4. Overview . . . . .	4
5. Structure of the MIB Module . . . . .	4
5.1. Relationship to the SNMPv2-MIB . . . . .	4
5.2. Relationship to the IF-MIB . . . . .	4
5.3. Relationship to the IP TUNNEL MIB . . . . .	4
5.4. MIB modules required for IMPORTS . . . . .	5
6. Definitions . . . . .	5
7. Security Considerations . . . . .	9
8. IANA Considerations . . . . .	10
9. Contributors . . . . .	10
10. Acknowledgements . . . . .	10
11. References . . . . .	10
11.1. Normative References . . . . .	10
11.2. Informative References . . . . .	11
Appendix A. Change Log . . . . .	11
Appendix B. Open Issues . . . . .	11
Authors' Addresses . . . . .	12

## 1. Introduction

IP TUNNEL MIB[RFC4087] are used for managing tunnels of any type over IPv4 and IPv6 networks, including Generic Routing Encapsulation (GRE)[RFC1701,RFC1702], IP-in-IP[RFC2003], Minimal Encapsulation [RFC2004], Layer 2 Tunneling Protocol (L2TP) [RFC2661], Point-to-Point Tunneling Protocol (PPTP) [RFC2637], Layer 2 Forwarding (L2F) [RFC2341], UDP (e.g., [RFC1234]), Ascend Tunnel Management Protocol (ATMP) [RFC2107], and IPv6-in-IPv4 [RFC2893] tunnels, among others. Over the past several years, there has been a number of "tunneling" protocols specified by the IETF (see [RFC1241] for an early discussion of the model and examples). This document describes a Management Information Base (MIB) module used for managing tunnels of any type over IPv4 and IPv6 networks, including Generic Routing Encapsulation (GRE) [RFC1701,RFC1702], IP-in-IP [RFC2003], Minimal Encapsulation [RFC2004], Layer 2 Tunneling Protocol (L2TP) [RFC2661], Point-to-Point Tunneling Protocol (PPTP) [RFC2637], Layer 2 Forwarding (L2F) [RFC2341], UDP (e.g., [RFC1234]), Ascend Tunnel Management Protocol (ATMP) [RFC2107], and IPv6-in-IPv4 [RFC2893] tunnels, among others.

This documents describes how to support IPv6 Rapid Deployment (6rd) [RFC5969] and Mapping of Address and Port (MAP)[draft-ietf-softwire-map] in IP TUNNEL MIB.

## 2. The Internet-Standard Management Framework

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

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

## 3. 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].

#### 4. Overview

IP TUNNEL MIB [RFC4087] are using provisioning for tunnel protocol, but could not support 6rd [RFC5969] and MAP [draft-ietf-softwire-map] due to lack of parameters. But MAP [draft-ietf-softwire-map] has compativility with DS-Lite [RFC6333] and stateless NAT64 [RFC6145]. Therefore if TUNNEL MIB once supports 6rd [RFC5969] and MAP[draft-ietf-softwire-map],it could manage many type of modern tunnels such as 6rd [RFC5969], MAP-T/MAP-E, DS-Lite [RFC6333], and XLAT464 CLAT [draft-ietf-v6ops-464xlat].

#### 5. Structure of the MIB Module

The MIB module specified herein provides one way to manage the 6rd and MAP devices thorough SNMP.

##### 5.1. Relationship to the SNMPv2-MIB

The 'system' group in the SNMPv2-MIB [RFC3418] is defined as being mandatory for all systems, and the objects apply to the entity as a whole. The 'system' group provides identification of the management entity and certain other system-wide data. The SAMPLE-MIB does not duplicate those objects.

##### 5.2. Relationship to the IF-MIB

The Interface MIB [RFC2863] requires that any MIB module which is an adjunct of the Interface MIB clarify specific areas within the Interface MIB. These areas were intentionally left vague in the Interface MIB to avoid over constraining the MIB, thereby precluding management of certain media-types.

Section 4 of [RFC2863] enumerates several areas which a media-specific MIB must clarify. The implementor is referred to [RFC2863] in order to understand the general intent of these areas.

##### 5.3. Relationship to the IP TUNNEL MIB

The IP Tunnel MIB [RFC4087] contains objects common to all IP tunnels, including 6rd/MAP. Additionally, tunnel encapsulation specific MIB (like what is defined in this document) extend the IP tunnel MIB to further describe encapsulation specific information.

for example:

6rd case

6rd prefix, 6rd Prefix Length, IPv4Mask Length

MAP case

rule IPv6 prefix, rule IPv6 prefix Length, rule IPv4 prefix , rule IPv4 prefix length, EA-bit length, PSID

tunnel method, BR address, source addresss could use tunnelIfEntry.

```
TunnelIfEntry ::= SEQUENCE {
    tunnelIfLocalAddress      IpAddress,    -- deprecated
    tunnelIfRemoteAddress     IpAddress,    -- deprecated
    tunnelIfEncapsMethod      IANA_tunnelType,
    tunnelIfHopLimit          Integer32,
    tunnelIfSecurity          INTEGER,
    tunnelIfTOS               Integer32,
    tunnelIfFlowLabel         IPv6FlowLabelOrAny,
    tunnelIfAddressType       InetAddressType,
    tunnelIfLocalInetAddress  InetAddress,
    tunnelIfRemoteInetAddress InetAddress,
    tunnelIfEncapsLimit       Integer32
}
```

tunnelIfEncapsMethod must be sixRd(xx), MAPT(xx) and MAPE(xx).

tunnelIfRemoteInetAddress must be BR address for CE. When 6rd, it would be IPv4 address. When MAP-T and MAP-E, it would be IPv6 address. 0.0.0.0 :: would be used for BR. TunnelIfXEntry would use for another prameters .

#### 5.4. MIB modules required for IMPORTS

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

#### 6. Definitions

```
tunnelIfXTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF TunnelIfXEntry
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This table contains additional objects for the tunnel
        interface table."
    ::= { tunnel xx }
```

```
tunnelIfXEntry OBJECT-TYPE
    SYNTAX      TunnelIfXEntry
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "An entry containing additional information applicable to a
        particular tunnel interface."
    INDEX       { ifIndex }
    ::= { tunnelIfXTable 1 }
```

```
TunnelIfXEntry ::= SEQUENCE {
    SamPrex      InetAddress,
    SamLength    Integer32
    BasePrex     InetAddress,
    BaseLength   Integer32
    EAbit        Integer32
    PSID         Integer32
}
}
```

```
SamPrefix OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Stateless Address Mapping Prex IPv4 for MAP,IPv6 for 6rd"
    ::= { TunnelIfXEntry 1 }
```

```
SamLength OBJECT-TYPE
    SYNTAX      Integer32(0..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Stateless Address Mapping length IPv4(0-31) for MAP,IPv6(0-127) for 6rd"
    ::= { TunnelIfXEntry 2 }
```

```
BasePrefix OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "rule IPv6 prefix for MAP, IPv4 address for 6rd"
    ::= { TunnelIfXEntry 3 }
```

```
BaseLength OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-write
    STATUS      current
```

## DESCRIPTION

"rule IPv6 prefix for MAP, IPv4 address for 6rd"

:= { TunnelIfXEntry 4 }

EABit OBJECT-TYPE

SYNTAX Integer32(0..127)

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"rule IPv6 prefix length for MAP, IPv4MaskLength for 6rd"

:= { TunnelIfXEntry 5 }

PSID OBJECT-TYPE

SYNTAX Integer32(0..127)

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"EA bit for MAP,0 must be for 6rd"

:= { TunnelIfXEntry 6 }

END

tunnelIfXTable OBJECT-TYPE

SYNTAX SEQUENCE OF TunnelIfXEntry

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"This table contains additional objects for the tunnel interface table."

::= { tunnel xx }

tunnelIfXEntry OBJECT-TYPE

SYNTAX TunnelIfXEntry

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"An entry containing additional information applicable to a particular tunnel interface."

INDEX { ifIndex }

::= { tunnelIfXTable 1 }

TunnelIfXEntry ::= SEQUENCE {

SamPrex InetAddress,

SamLength Integer32

BasePrex InetAddress,



```
        BaseLength      Integer32
        EAbit           Integer32
        PSID            Integer32
    }
}

SamPrefix OBJECT-TYPE
SYNTAX      InetAddress
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"Stateless Address Mapping Prex IPv4 for MAP,IPv6 for 6rd"
:= { TunnelIfXEntry 1 }

SamLength OBJECT-TYPE
SYNTAX      Integer32(0..127)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"Stateless Address Mapping length IPv4(0-31) for MAP,IPv6(0-127) for 6rd"
:= { TunnelIfXEntry 2 }

BasePrefix OBJECT-TYPE
SYNTAX      InetAddress
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"rule IPv6 prefix for MAP, IPv4 address for 6rd"
:= { TunnelIfXEntry 3 }

BaseLength OBJECT-TYPE
SYNTAX      InetAddress
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"rule IPv6 prefix for MAP, IPv4 address for 6rd"
:= { TunnelIfXEntry 4 }

EAbit      OBJECT-TYPE
SYNTAX      Integer32(0..127)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"rule IPv6 prefix length for MAP, IPv4MaskLength for 6rd"
:= { TunnelIfXEntry 5 }

PSID      OBJECT-TYPE
SYNTAX      Integer32(0..127)
```

```
MAX-ACCESS read-write
STATUS      current
DESCRIPTION
"EA bit for MAP,0 must be for 6rd"
:= { TunnelIfXEntry 6 }
```

END

## 7. Security Considerations

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

There are no management objects defined in this MIB module that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB module is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB module via direct SNMP SET operations.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

- o 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.

## 8. IANA Considerations

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

Descriptor	OBJECT IDENTIFIER value
-----	-----
TunnelIXEntry	{ tunnel XXX }
IANAtunnelType	::= TEXTUAL-CONVENTION
SYNTAX	INTEGER {
	sixRd ("XX")           -- 6rd encapsulation
	MAPT ("XX")           -- MAP-T encapsulation
	MAPE ("XX")           -- MAP-T encapsulation
	}

## 9. Contributors

This template is based on contributions from the Mib Doctors, especially Juergen Schoenwaelder, Dave Perkins, C.M.Heard and Randy Presuhn.

## 10. Acknowledgements

Thanks to Marshall Rose for developing the XML2RFC format.

## 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.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information

Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.
- [RFC4181] Heard, C., "Guidelines for Authors and Reviewers of MIB Documents", BCP 111, RFC 4181, September 2005.

#### 11.2. Informative References

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

#### Appendix A. Change Log

The following changes have been made from draft-xxx-xxx-xxx-12 .

[TODO] replace this list with your own list

1. Updated the introductory boilerplate text, the security considerations section and the references to comply with the current IETF standards and guidelines.
2. Additions and clarifications in various description clauses.

#### Appendix B. Open Issues

[TODO] This list of open issues should be cleared and removed before this document hits the IESG.

1. Contributor addresses need to be updated

#### Authors' Addresses

Shishio Tsuchiya (editor)  
Cisco Systems  
Midtown Tower, 9-7-1, Akasaka  
Minato-Ku, Tokyo 107-6227  
Japan

Phone: +81 3 6434 6543  
Email: [shtsuchi@cisco.com](mailto:shtsuchi@cisco.com)

Jacni Qin  
Cisco Systems  
Shanghai  
China

Phone:  
Email: [jacni@jacni.com](mailto:jacni@jacni.com)



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 17, 2013

Q. Sun  
C. Xie  
China Telecom  
Y. Lee  
Comcast  
M. Chen  
FreeBit  
July 16, 2012

Deployment Considerations for Lightweight 4over6  
draft-sun-softwire-lightweigh-4over6-deployment-02

Abstract

Lightweight 4over6 is a mechanism which moves the translation function from tunnel Concentrator (AFTR) to Initiators (B4s), and hence reduces the mapping scale on the Concentrator to per-customer level. This document discusses various deployment models of Lightweight 4over6. It also describes the deployment considerations and applicability of the Lightweight 4over6 architecture.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 17, 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. Requirements Language . . . . .	4
3. Deployment Model . . . . .	5
4. Overall Deployment Considerations . . . . .	7
4.1. Addressing and Routing . . . . .	7
4.2. Port-set Management . . . . .	7
4.3. Concentrator Discovery . . . . .	7
5. Concentrator Deployment Consideration . . . . .	9
5.1. Logging at the Concentrator . . . . .	9
5.2. Reliability Considerations of Concentrator . . . . .	9
5.3. Placement of AFTR . . . . .	9
5.4. Port set algorithm consideration . . . . .	10
6. DS-Lite Compatibility . . . . .	11
6.1. Case 1: Integrated Network Element with Lightweight 4over6 and DS-Lite AFTR Scenario . . . . .	11
6.2. Case 2: DS-Lite Coexistent scenario with Separated AFTR . . . . .	12
7. Acknowledgement . . . . .	13
8. References . . . . .	14
Appendix 1. Appendix:Experimental Result . . . . .	16
1.1. Experimental environment . . . . .	16
1.2. Experimental results . . . . .	17
1.3. Conclusions . . . . .	18
Authors' Addresses . . . . .	19



## 1. Introduction

Lightweight 4over6 [I-D.cui-software-b4-translated-ds-lite] is an extension to DS-Lite which simplifies the AFTR module [RFC6333] with distributed NAT function among B4 elements. The Initiator in Lightweight 4over6 is provisioned with an IPv6 address, an IPv4 address and a port-set. It performs NAPT on end user's packets with the provisioned IPv4 address and port-set. IPv4 packets are forwarded between the Initiator and the Concentrator over a Software using IPv4-in-IPv6 encapsulation. The Concentrator maintains one mapping entry per subscriber with the IPv6 address, IPv4 address and port-set. Therefore, this extension removes the NAT44 module from the AFTR and replaces the session-based NAT table to a per-subscriber based mapping table. This should relax the requirement to create dynamic session-based log entries. This mechanism preserves the dynamic feature of IPv4/IPv6 address binding as in DS-Lite, so it has no coupling between IPv6 address and IPv4 address/port-set as any full stateless solution ([RFC6052] or [I-D.ietf-software-map]) requires. This document discusses deployment models of Lightweight 4over6. It also describes the deployment considerations and applicability of the Lightweight 4over6 architecture.

Terminology of this document follows the definitions and abbreviations of [I-D.cui-software-b4-translated-ds-lite].

## 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. Deployment Model

Lightweight 4over6 is suitable for operators who would like to free any correlation of the IPv6 address with IPv4 address and port-set (or port-range). In comparison to full stateless solutions like MAP [I-D.ietf-software-map] and 4rd [I-D.ietf-software-4rd], Lightweight 4over6 frees address planning of IPv6 delegation for CPE from mapping rule administration and management in the network. Thus, IPv6 addressing is completely flexible to fit other deployment requirements, e.g., auto-configuration, service classification, user management, QoS support, etc. The philosophy here is that bits of IPv6 address should be left for IPv6 usage first.

Lightweight 4over6 can be deployed in a residential network (depicted in Figure1). In this scenario, an Initiator would acquire an IPv4 address and a port-set after a successful user authentication process and IPv6 provisioning process. Then, it establishes an IPv4-in-IPv6 software using the IPv6 address to deliver IPv4 services to its connected host via the Concentrator in the network. The Initiator can act as a CPE, or software located in the host. The Concentrator supports Lightweight 4over6 which keeps the mapping between Initiator's IPv6 address and its allocated IPv4 address + port set. The supporting server may keep the binding information as well for logging and user management.

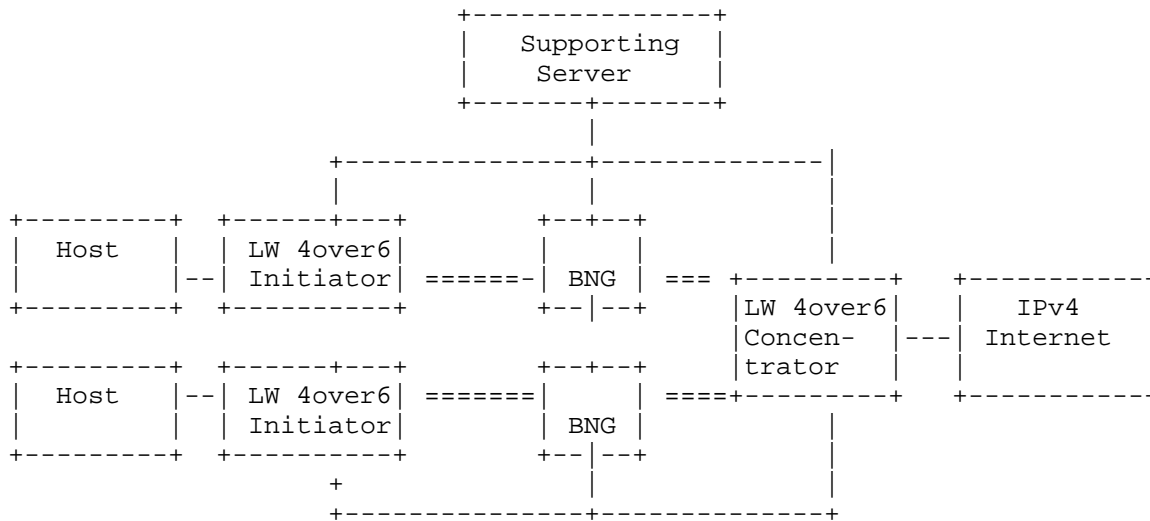


Figure 1 Deployment Model

There are two deployment models in practice: one is called bottom-up and the other is top-down. In bottom-up model, after port-restricted

IPv4 address is allocated to a given subscriber, the Concentrator will report mapping records to the support server on creating a binding for traffic logging if necessary. In this way, the Concentrator can determine the binding by its own and there is little impact on existing network architecture. In top-down model, the Supporting system should firstly determine the binding information for each subscriber and then synchronize it with the Concentrator. With this method, one binding record can be easily synchronized with multiple Concentrators and stateless failover can be achieved. However, new mechanism (e.g. Netconf) needs to be introduced to notify each individual binding record between the Supporting system and the Concentrator.

## 4. Overall Deployment Considerations

### 4.1. Addressing and Routing

In Lightweight 4over6, there is no inter-dependency between IPv4 and IPv6 addressing schemes. IPv4 address pools are configured centralized in Concentrator for IPv6 subscribers. These IPv4 prefix must advertise to IPv4 Internet accordingly.

For IPv6 addressing and routing, there are no additional addressing and routing requirements. The existing IPv6 address assignment and routing announcement should not be affected. For example, in PPPoE scenario, a CPE could obtain a prefix via prefix delegation procedure, and the hosts behind CPE would get its own IPv6 addresses within the prefix through SLAAC or DHCPv6 statefully. This IPv6 address assignment procedure has nothing to do with restricted IPv4 address allocation.

### 4.2. Port-set Management

In Lightweight 4over6, each Initiator will get its restricted IPv4 address and a valid port-set after successful user authentication process and IPv6 provisioning process. This port-set assignment should be synchronized between port management server and the Concentrator. The port management server is responsible for allocating port restricted IPv4 address to the Initiator. It can be new option to the DHCPv4 server [I-D.bajko-pripaddrassign]. The DHCPv4 server can either be collocated in the Concentrator or a dedicated server.

Different mechanisms including PCP- extended protocol [I-D.tsou-pcp-natcoord], DHCP-extended protocol or IPCP-extended protocol, etc., can also be used.

Compared with DHCP-based mechanism, PCP-based mechanism is more flexible. An Initiator can send multiple PCP requests simultaneously to acquire a number of ports or use [I-D.tsou-pcp-natcoord] for one-time port-set allocation.

### 4.3. Concentrator Discovery

A Lightweight 4over6 Initiator must discover the Concentrator's IPv6 address before offering any IPv4 services. This IPv6 address can be learned through an out-of-band channel, static configuration, or dynamic configuration. In practice, Lightweight 4over6 Initiator can use the same DHCPv6 option [RFC6334] to discover the FQDN of the Concentrator. When Lightweight 4over6 is deployment in the same place with DS-Lite, different FQDNs can be configured for Lightweight

4over6 and DS-Lite separately ( More detailed consideration on DS-Lite compatibility will be discussed in Section...).

## 5. Concentrator Deployment Consideration

As Lightweight 4over6 is an extension to DS-Lite, both technologies share similar deployment considerations. For example: Interface consideration, MTU, Fragment, Lawful Intercept Considerations, Blacklisting a shared IPv4 Address, AFTR's Policies, AFTR Impacts on Accounting Process, etc., in [I-D.ietf-softwire-dslite-deployment] can also be applied here. This document only discusses new considerations specific to Lightweight 4over6.

### 5.1. Logging at the Concentrator

In Lightweight 4over6, operators only log one entry per subscriber. The log should include subscriber's IPv6 address used for the softwire, the public IPv4 address and the port-set. The port set algorithm implemented in Lightweight 4over6 Concentrator should be synchronized with the one implemented in logging system. For example, if contiguous port set algorithm is adopted in the Concentrator, the same algorithm should also be applied to the logging system.

### 5.2. Reliability Considerations of Concentrator

In Lightweight 4over6, subscriber to IPv4 and port-set mapping must be pre-provisioned in the Concentrator before providing IPv4 services. For redundancy, the backup Concentrator must either have the subscriber mapping already provisioned or notify the Initiator to create a new mapping in the backup Concentrator. The first option can be considered as hot standby mode. The second option may require a new notification mechanism which is outside the scope of this document.

### 5.3. Placement of AFTR

The Concentrator can be deployed in a "centralized model" or a "distributed model".

In the "centralized model", the Concentrator could be located at the higher place, e.g. at the exit of MAN, etc. Since the Concentrator has good scalability and can handle numerous concurrent sessions, we recommend to adopt the "centralized model" for Lightweight 4over6 as it is cost-effective and easy to manage.

In the "distributed model", Concentrator is usually integrated with the BRAS/SR. Since newly emerging customers might be distributed in the whole Metro area, we have to deploy Concentrator on all BRAS/SRs. This will cost a lot in the initial phase of the IPv6 transition period.

#### 5.4. Port set algorithm consideration

If each Initiator is given a set of ports, port randomization algorithm can only select port in the given port-set. This may introduce security risk because hackers can make a more predictable guess of what port a subscriber may use. Therefore, non-continuous port set algorithms (e.g. as defined in [I-D.ietf-softwire-map]) can be used to improve security.



## 6. DS-Lite Compatibility

Lightweight 4over6 can be either deployed all alone, or combined with DS-Lite [RFC6333]. Since Lightweight 4over6 does not any have extra requirement on IPv6 addressing, it can use the same addressing scheme with DS-Lite, together with routing policy, user management policy, etc. Besides, the bottom-up model has quite similar requirement and workflow on the support server with DS-Lite. Therefore, it is suitable for operators to deploy incrementally in existing DS-Lite network

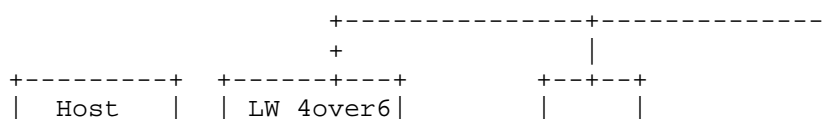
### 6.1. Case 1: Integrated Network Element with Lightweight 4over6 and DS-Lite AFTR Scenario

In this case, DS-Lite has been deployed in the network. Later in the deployment schedule, the operator decided to implement Lightweight 4over6 Concentrator function in the same network element (depicted in Figure2). Therefore, the same network element needs to support both transition mechanisms.

There are two options to distinguish the traffic from two transition mechanisms.

The first one is to distinguish using the client's source IPv4 address. The IPv4 address from Lightweight 4over6 is public address as NAT has been done in the Initiator, and IPv4 address for DS-lite is private address as NAT will be done on AFTR. When the network element receives an encapsulated packet, it would de-capsulate packet and apply the transition mechanism based on the IPv4 source address in the packet. This requires the network element to examine every packet and may introduce significant extra load to the network element. However, both the B4 element and Lightweight 4over6 Initiator can use the same DHCPv6 option [RFC6334] with the same FQDN of the AFTR and Concentrator.

The second one is to distinguish using the destination's tunnel IPv6 address. One network element can run separated instances for Lightweight 4over6 and DS-Lite with different tunnel addresses. Then B4 element and Lightweight 4over6 Initiator can use the same DHCPv6 option [RFC6334] with different FQDNs pointing to corresponding tunnel addresses. This requires the support server should distinguish different types of users when assigning the FQDNs in DHCPv6 process.



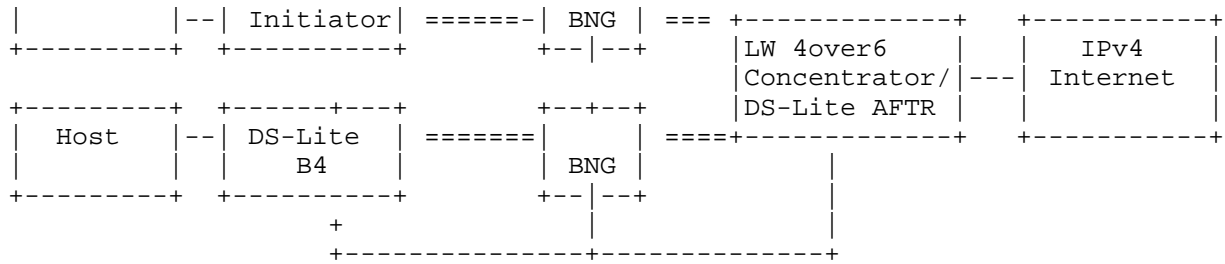


Figure 2 DS-Lite Coexistence scenario with Integrated AFTR

## 6.2. Case 2: DS-Lite Coexistent scenario with Separated AFTR

This is similar to Case 1. The difference is the Concentrator and AFTR functions won't be co-located in the same network element (depicted in Figure3). This use case decouples the functions to allow more flexible deployment. For example, an operator may deploy AFTR closer to the edge and Concentrator closer to the core. Moreover, it does not require the network element to pre-configure with the CPE's IPv6 addresses. An operator can deploy more AFTR and Concentrator at needed. However, this requires the B4 and Initiator to discover the corresponding network element. In this case, B4 element and Lightweight 4over6 Initiator can still use [RFC6334] with different FQDNs pointing to corresponding tunnel end-point addresses, and the support server should distinguish different types of users.

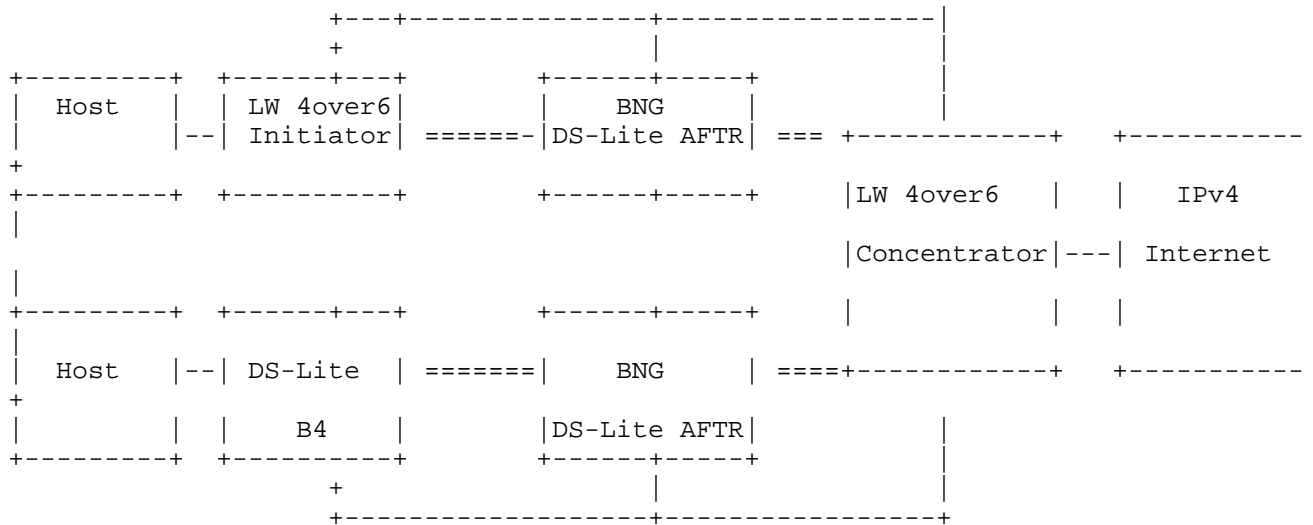


Figure 3 DS-Lite Coexistence scenario with Separated AFTR

## 7. Acknowledgement

TBD

## 8. References

- [I-D.bajko-pripaddrassign]  
Bajko, G., Savolainen, T., Boucadair, M., and P. Levis,  
"Port Restricted IP Address Assignment",  
draft-bajko-pripaddrassign-04 (work in progress),  
April 2012.
- [I-D.bsd-software-stateless-port-index-analysis]  
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-dhc-dhcpv4-over-ipv6]  
Cui, Y., Wu, P., Wu, J., and T. Lemon, "DHCPv4 over IPv6  
transport", draft-cui-dhc-dhcpv4-over-ipv6-00 (work in  
progress), October 2011.
- [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-07  
(work in progress), July 2012.
- [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.ietf-dhc-dhcpv4-over-ipv6]  
Cui, Y., Wu, P., Wu, J., and T. Lemon, "DHCPv4 over IPv6  
Transport", draft-ietf-dhc-dhcpv4-over-ipv6-03 (work in  
progress), May 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-26 (work in progress), June 2012.
- [I-D.ietf-software-4rd]  
Despres, R., Penno, R., Lee, Y., Chen, G., and S. Jiang,  
"IPv4 Residual Deployment via IPv6 - a unified Stateless  
Solution (4rd)", draft-ietf-software-4rd-02 (work in  
progress), June 2012.
- [I-D.ietf-software-dslite-deployment]

Lee, Y., Maglione, R., Williams, C., Jacquenet, C., and M. Boucadair, "Deployment Considerations for Dual-Stack Lite", draft-ietf-softwire-dslite-deployment-03 (work in progress), March 2012.

[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.

[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.sun-v6ops-laft6]

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

[I-D.tsou-pcp-natcoord]

Sun, Q., Boucadair, M., Deng, X., Zhou, C., and T. Tsou, "Using PCP To Coordinate Between the CGN and Home Gateway Via Port Allocation", draft-tsou-pcp-natcoord-05 (work in progress), March 2012.

[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.

[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.

[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.

## 1. Appendix:Experimental Result

We have deployed Lightweight 4over6 in our operational network of HuNan province, China. It is designed for broadband access network, and different versions of Initiator have been implemented including a linksys box, a software client for windows XP, vista and Windows 7. It can be integrated with existing dial-up mechanisms such as PPPoE, etc. The major objectives listed below aimed to verify the functionality and performance of Lightweight 4over6:

- o Verify how to deploy Lightweight 4over6 in a practical network.
- o Verify the impact of applications with Lightweight 4over6.
- o Verify the performance of Lightweight 4over6.

### 1.1. Experimental environment

The network topology for this experiment is depicted in Figure 2.

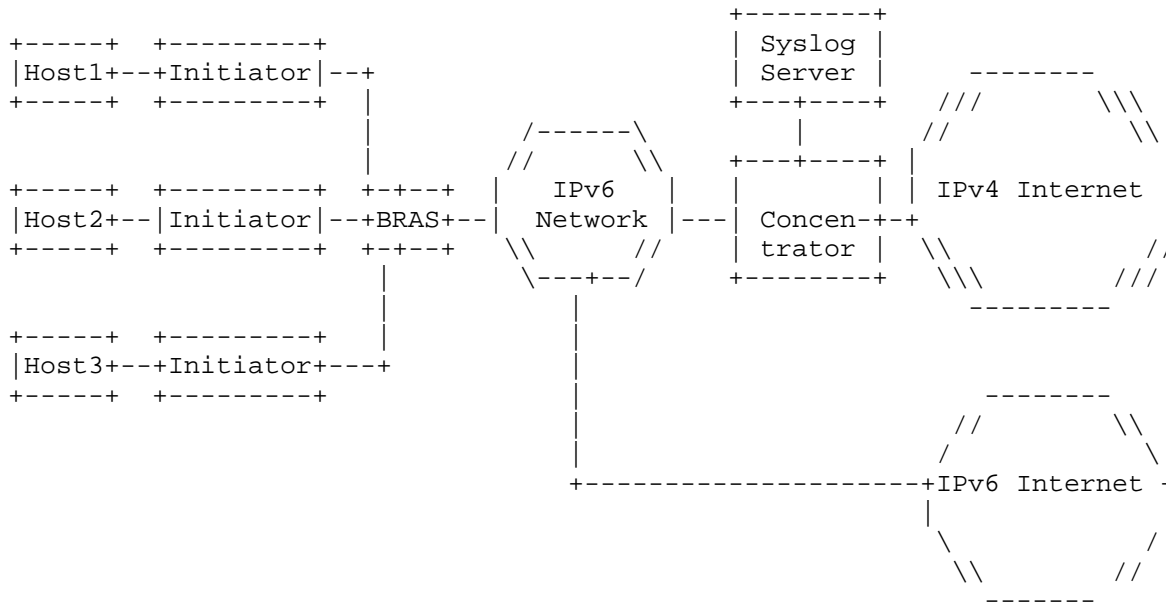


Figure 2 Lightweight 4over6 experiment topology

In this deployment model, Concentrator is co-located with a extended PCP server to assign restricted IPv4 address and port set for Initiator. It also triggers subscriber-based logging event to a centrilized syslog server. IPv6 address pools for subscribers have

been distributed to BRASs for configuration, while the public available IPv4 address pools are configured by the centralized Concentrator with a default address sharing ratio. It is rather flexible for IPv6 addressing and routing, and there is little impact on existing IPv6 architecture.

In our experiment, Initiator will firstly get its IPv6 address and delegated prefix through PPPoE, and then initiate a PCP-extended request to get public IPv4 address and its valid port set. The Concentrator will thus create a subscriber-based state accordingly, and notify syslog server with {IPv6 address, IPv4 address, port set, timestamp}.

## 1.2. Experimental results

In our trial, we mainly focused on application test and performance test. The applications have widely include web, email, Instant Message, ftp, telnet, SSH, video, Video Camera, P2P, online game, voip and so on. For performance test, we have measured the parameters of concurrent session numbers and throughput performance.

The experimental results are listed as follows:

Application Type	Test Result	Port Number Occupation
Web	ok IE, Firefox, Chrome	normal websites: 10~20 Ajex Flash webs: 30~40
Video	ok, web based or client based	30~40
Instant Message	ok QQ, MSN, gtalk, skype	8~20
P2P	ok utorrent,emule,xunlei	lower speed: 20~600 (per seed) higher speed: 150~300
FTP	need ALG for active mode, flashxp	2
SSH, TELNET	ok	1 for SSH, 3 for telnet
online game	ok for QQ, flash game	20~40

Figure 3 Lightweight 4over6 experimental result

The performance test for Concentrator is taken on a normal PC. Due to limitations of the PC hardware, the overall throughput is limited to around 800 Mbps. However, it can still support more than one hundred million concurrent sessions.

### 1.3. Conclusions

From the experiment, we can have the following conclusions:

- o Lightweight 4over6 has good scalability. As it is a lightweight solution which only maintains per-subscription state information, it can easily support a large amount of concurrent subscribers.
- o Lightweight 4over6 can be deployed rapidly. There is no modification to existing addressing and routing system in our operational network. And it is simple to achieve traffic logging.
- o Lightweight 4over6 can support a majority of current IPv4 applications.



## Authors' Addresses

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

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

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

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

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

Email: yiu\_lee@cable.comcast.com

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

Email: fibrib@gmail.com



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: December 29, 2012

T. Tsou, Ed.  
Huawei Technologies (USA)  
B. Li  
Huawei Technologies  
J. Schoenwaelder  
Jacobs University Bremen  
R. Penno  
Cisco Systems, Inc.  
June 27, 2012

DS-Lite Failure Detection and Failover  
draft-tsou-softwire-bfd-ds-lite-03

Abstract

In DS-Lite, the tunnel is stateless, not associated with any state information, and no failure detection and failover mechanism is available. This makes it difficult to manage and diagnose if there is a problem. This draft analyzes the applicability of some of the possible solutions.

Status of this Memo

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

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

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

This Internet-Draft will expire on December 29, 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. Solutions . . . . .	3
3.1. Bidirectional Forwarding Detection (BFD) . . . . .	3
3.1.1. DS-Lite Scenario . . . . .	4
3.1.2. Parameters for BFD . . . . .	4
3.1.3. Elements of Procedure . . . . .	5
3.1.4. Implementation Considerations . . . . .	5
3.2. Port Control Protocol (PCP) . . . . .	6
3.3. ICMP Echo (Request) / Echo Reply (PING) . . . . .	6
4. Failover . . . . .	7
5. IANA Considerations . . . . .	7
6. Security Considerations . . . . .	7
7. Acknowledgements . . . . .	7
8. References . . . . .	7
8.1. Normative References . . . . .	7
8.2. Informative References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

In DS-Lite [RFC6333], the IPv4-in-IPv6 DS-Lite tunnel is stateless, no status information about the tunnel is available, and no keep-alive mechanism is available. It is difficult to know whether the tunnel is up or down; and if there is a link problem, the Basic Bridging BroadBand (B4) element can not automatically switch to another Address Family Transition Router (AFTR) so as to continue the network service automatically, without the involvement of operators. This lack of failure detection and failover creates problems for network operation and maintenance.

Possible solutions for failure detection include the usage of Bidirectional Forwarding Detection (BFD), the Port Control Protocol (PCP), and ICMP Echo (Request) / Echo Reply (PING). The properties of these solutions are discussed in this document and guidelines are provided how to implement failure detection and automatic failover.

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

AFTR: Address Family Transition Router.

B4: Basic Bridging BroadBand.

BBF: BroadBand Forum.

BFD: Bidirectional Forwarding Detection.

CPE: Customer Premise Equipment (i.e., the DS-Lite B4).

FQDN Fully Qualified Domain Name.

PCP Port Control Protocol.

## 3. Solutions

### 3.1. Bidirectional Forwarding Detection (BFD)

Bidirectional Forwarding Detection [RFC5880] (BFD) is a mechanism intended to detect faults in a bidirectional path. It is usually used in conjunction with applications like OSPF, IS-IS, for fast fault recovery and fast re-route [RFC5882]. BFD is being made

mandatory for keep-alive for subscriber sessions, including DS-Lite, by the BroadBand Forum (BBF) [WT-146].

BFD can be used in DS-Lite, by creating a BFD session between the B4 element and the AFTR to provide tunnel status information. If a fault is detected, the B4 element can try to create a DS-Lite tunnel with another AFTR and terminate the existing one, so as to continue network service.

[I-D.vinokour-bfd-dhcp] proposes using a DHCP option to distribute BFD parameters to B4 elements. But in case of DS-Lite, some of the key BFD parameters are already available (e.g., peer IP address), and other parameters can be negotiated by BFD signaling or statically configured, so that no extra DHCP option(s) need to be defined.

### 3.1.1. DS-Lite Scenario

In DS-Lite [RFC6333], the BFD packet SHOULD be sent through an IPv4-in-IPv6 tunnel, as shown in Figure 1. The IPv4 addresses of the B4 element and the AFTR SHOULD be the endpoints of a BFD session.

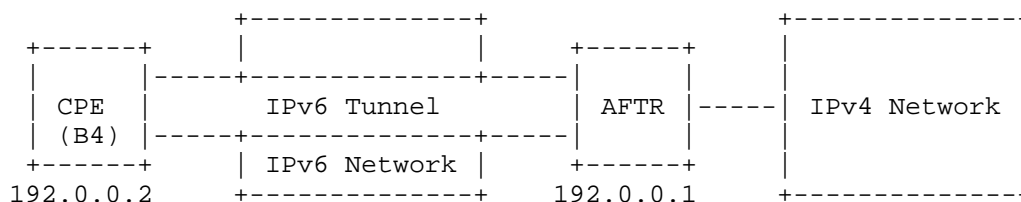


Figure 1: DS-Lite Scenario

### 3.1.2. Parameters for BFD

In order to set up a BFD session, the following parameters are needed, as shown in Section 4.1 of [RFC5880]:

- o Peer IP address
- o My Discriminator
- o Your Discriminator
- o Desired Min TX Interval
- o Required Min RX Interval
- o Required Min Echo RX Interval

In DS-Lite [RFC6334], the B4's WAN-side IPv4 address is the well-known address 192.0.0.2, and the AFTR's well-known IPv4 address is 192.0.0.1, as defined in section 5.7 of [RFC6333]. The B4 element needs to create an IPv6 tunnel to an AFTR so as to get network connectivity to the AFTR, and send IPv4 BFD packets through the tunnel to manage it.

The other parameters listed above can be negotiated by BFD signaling, and initial values can be configured on B4 elements and AFTRs.

### 3.1.3. Elements of Procedure

When a B4 element gets online, it will be assigned an IPv6 prefix or address, and also the FQDN of the AFTR, as defined in [RFC6334]. The B4 element will create an IPv6 tunnel to the AFTR with which the B4 element can initiate a BFD session to the AFTR. BFD packets will be sent through the DS-Lite tunnel. As defined in section 4 of [RFC5881], BFD control packets MUST be sent in UDP packets with destination port 3784, and BFD echo packets MUST be sent in UDP packets with destination port 3785.

When sending out the first BFD packet, the B4 element can generate a unique local discriminator, and set the remote discriminator to zero. When the AFTR receives the first BFD packet from a B4 element, the AFTR will also generate a corresponding local discriminator, and put it in the response packet to the B4 element. This will finish the discriminator negotiation in the B4 to AFTR direction, without any manual configuration.

When an AFTR receives the first packet from a B4 element, the AFTR will get the IPv6 address and discriminator of the B4 element, so that the AFTR can initiate the BFD session in the other direction and a similar discriminator negotiation can be carried out.

### 3.1.4. Implementation Considerations

BFD is usually used for quick fault detection, at a very small time scale, e.g. milliseconds. But in DS-Lite, it may not be necessary to detect faults in such a short time. On the other hand, an AFTR may need to support tens of thousands of B4 elements, which means an AFTR will need to support the same number of BFD sessions. In order to meet performance requirements on an AFTR, it may be necessary to extend the time period between BFD packet transmissions to a longer time, e.g., 10s or 30s.

Compared to other solutions, BFD has a simple and fixed packet format, which is easy to implement by logic devices (e.g., ASIC, FPGA). Complicated protocols are usually processed by software which

is relatively slow. An AFTR may need to support 10000-20000 users, and if the protocol is handled by software, it will bring extra load to the AFTR.

### 3.2. Port Control Protocol (PCP)

PCP [I-D.ietf-pcp-base-26] is a NAT traversal tool. It can also be used for network connectivity test if PCP is supported in the network. A common use case of PCP is to create a pinhole so that external users can visit the servers located behind a NAT. The lifetime of the pinhole mapping is usually long, e.g., hours, and the lifetime will be refreshed periodically by the client before it is expired. For the purpose of network connectivity tests, a B4 element can create a mapping in the CGN via PCP, with a short life time, e.g., 10s of seconds, and keep on refreshing the mapping before it expires. If any refresh requests fail, the B4 element knows that something is wrong with the link or the PCP server or the CGN.

In order to detect the network connectivity of the DS-Lite tunnel, the encapsulation mode MUST be used for PCP: PCP packets are sent through the DS-Lite tunnel. Encapsulation mode and plain mode are two alternatives for PCP, there is no consensus yet which one should be preferred in the PCP specification.

PCP can detect the failure of more components of the DS-Lite system. Besides failures of the link and the routing, it also covers NAT functions.

### 3.3. ICMP Echo (Request) / Echo Reply (PING)

PING is commonly implemented using the Echo (Request) and Echo Response messages of the Internet Control Message Protocol (ICMP) [RFC0792] [RFC4443]. In case of DS-Lite, a B4 element can send Echo (Request) packets to the AFTR periodically. If the B4 element does not receive Echo Response packets for a certain number (e.g., 3) of Echo (Request) packets, then the B4 element decides that a fault has been detected.

In order to test the connectivity of DS-Lite tunnel, Echo (Request) packets MUST be sent using ICMPv4, rather than ICMPv6.

Since ICMP is an integral part of any IP implementation, the usage of PING to detect tunnel failures does not require any special implementation efforts on the B4 elements. However, on AFTRs that process ICMP messages in software rather than in hardware, the usage of PING might lead to scalability issues.



#### 4. Failover

The FQDN of the AFTR is sent to the B4 element via a DHCP option, as defined in [RFC6334]. Multiple IP addresses can be configured for the FQDN of an AFTR on the DNS server. If a B4 element detects a failure on the link to the AFTR, the B4 element MUST terminate the current DS-Lite tunnel, choose another AFTR address in the list, and create a tunnel to the new AFTR. If necessary, the B4 element SHOULD re-configure the connectivity test tool accordingly and restart the test procedures.

Anycasts may also be used for failover. But there is an ICMP-error-message problem with anycast, that is, when a packet is sent from the AFTR to a B4 element, if one of the routers along the path generates an ICMP error message, e.g., Packet Too Big (PTB), then the error message may not be sent back to the source AFTR but to another AFTR.

#### 5. IANA Considerations

This memo includes no request to IANA.

#### 6. Security Considerations

In the DS-Lite [RFC6333] application, the B4 element may not be directly connected to the AFTR; there may be other routers between them. In such a deployment, there are potential spoofing problems, as described in [RFC5883]. Hence cryptographic authentication SHOULD be used with BFD as described in [RFC5880] if security is concerned.

#### 7. Acknowledgements

The authors would like to thank Mohamed Boucadair for his useful comments.

#### 8. References

##### 8.1. Normative References

[I-D.ietf-pcp-base-26]

Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)(work in progress)", Jun 2012.

[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.
- [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.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, June 2010.
- [RFC5882] Katz, D. and D. Ward, "Generic Application of Bidirectional Forwarding Detection (BFD)", RFC 5882, June 2010.
- [RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, June 2010.
- [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.
- [WT-146] Kavanagh, A., Klammm, F., Boucadair, W., and R. Dec, "WT-146 Subscriber Sessions (work in progress)", Apr 2012.

## 8.2. Informative References

- [I-D.vinokour-bfd-dhcp]  
Vinokour, V., "Configuring BFD with DHCP and Other Musings", May 2008.

Authors' Addresses

Tina Tsou (editor)  
Huawei Technologies (USA)  
2330 Central Expressway  
Santa Clara CA 95050  
USA

Phone: +1 408 330 4424  
Email: tina.tsou.zouting@huawei.com

Brandon Li  
Huawei Technologies  
M6, No. 156, Beiqing Road, Haidian District  
Beijing 100094  
China

Phone:  
Email: brandon.lijian@huawei.com

Juergen Schoenwaelder  
Jacobs University Bremen  
Campus Ring 1  
Bremen 28759  
Germany

Phone:  
Email: j.schoenwaelder@jacobs-university.de

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

Phone:  
Email: repenno@cisco.com



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: January 17, 2013

T. Tsou, Ed.  
Huawei Technologies (USA)  
T. Murakami  
IP Infusion  
S. Perreault  
Viagenie  
July 16, 2012

Port Set Definition Algorithms Analysis  
draft-tsou-softwire-port-set-algorithms-analysis-02

Abstract

This memo analyses the some port set definition algorithms which encodes port set information into IPv6 address so as to support stateless IPv4 to IPv6 transition technologies, e.g. 4rd-U and 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 17, 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. Terminology . . . . .	4
3. Various types of algorithms . . . . .	4
3.1. GMA style algorithms . . . . .	4
3.1.1. MAP . . . . .	4
3.1.2. 4rd-U . . . . .	6
3.1.3. Summary . . . . .	7
3.2. Mask/Value style algorithms . . . . .	7
3.3. Cryptographical style algorithms . . . . .	9
4. Conclusion . . . . .	10
5. IANA Considerations . . . . .	10
6. Security Considerations . . . . .	10
7. References . . . . .	10
7.1. Normative References . . . . .	10
7.2. Informative References . . . . .	11
Authors' Addresses . . . . .	11

## 1. Introduction

Some stateless IPv4 to IPv6 stransition technologies are invented by the industrial to provide IPv4 network service through IPv6 network, which also support IPv4 address sharing via port sets. These technologies can significantly simplify the implementation of the border router and reduce resource requirement.

In these solutions, a port set is assigned to each CPE, and can be calculated by a port set ID in conjunction with some other parameters; for any port number, the corresponding port set ID can also be derived, that means, the mapping algorithm must be reversible. When the CPE needs to send an IPv4 packet, it can map an IPv4 packet into an IPv6 packet, either by translation or encapsulation, the IPv4 address and port set ID will be embedded into an IPv6 address; when the BR receive the IPv6 packet, it will decapsulate it. When the BR need to forward an IPv4 packet to the CPE, it will first derive the port set ID from the port, and then map the IPv4 packet into an IPv6 packet.

In order to support these technologies, some port set definition algorithms are worked out. It may be useful to analyse the characteristics of these algorithms for better understanding and to choose a proper algorithm for different needs.

A good port set definition algorithm must be reversible, easy to implement, and should be able to define non-continuous or random port sets for better security, be able to exclude the well known ports, 0 ~ 1023 or 0 ~ 4095, etc.

This memo will analyse the following characteristics:

- o Port set type: continuous, non-continuous, random
- o Stateless: yes or no
- o Security: security level, continuous port set provides common security, random port set provides good security.
- o Implementation: implementation complexity, performance, etc.
- o Friendliness for NAT44: comply with NAT44 or not
- o Sharing ratio: maximum, minimum sharing ratio
- o Revert calculation from port number to PSID at BR.

- o Exclude well known ports

## 2. Terminology

BR: Border Router.

CPE: Customer Premise Equipment.

GMA: Generalized Modulus Algorithm.

MAP: Map Address and Port.

PSID: Port Set ID, one of the key parameters used to derived a set of ports.

## 3. Various types of algorithms

Currently, the port set definition algorithms can be classified into three categories: GMA style, Mask/Value style and cryptographical style.

### 3.1. GMA style algorithms

Currently there are three sets of draft support GMA style algorithm: MAP [I-D.ietf-softwire-map-01], 4rd-U [I-D.ietf-softwire-4rd-02] and, but they are not exactly all the same.

#### 3.1.1. MAP

In MAP [I-D.ietf-softwire-map-01], a port set can be defined by the following parameters:

R: sharing ratio;

P: PSID;

M: maximum number of contiguous ports.

To derive a port from the port set, the following equation can be used:

$$\text{Port} = R * M * j + M * P + i$$

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



$i$  is the port index in a sub port set,  $i = 0$  to  $M-1$ ;

To derive the PSID from a given port:

$PSID = (\text{floor}(\text{Port}/M)) \% R$ , where  $\%$  is the modulus operator.

Parameter  $M$  is to generate non-continuous ports sets, rather than a single continuous port set, which brings better security. If  $M=1$ , a single continuous port set is defined.

PSID will be encoded in the IPv6 address, as shown in Figure 1 and Figure 2.

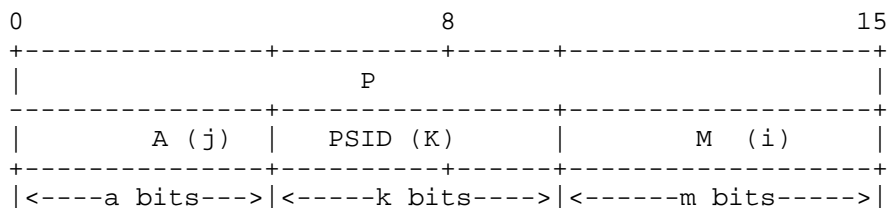


Figure 1: Bit representation

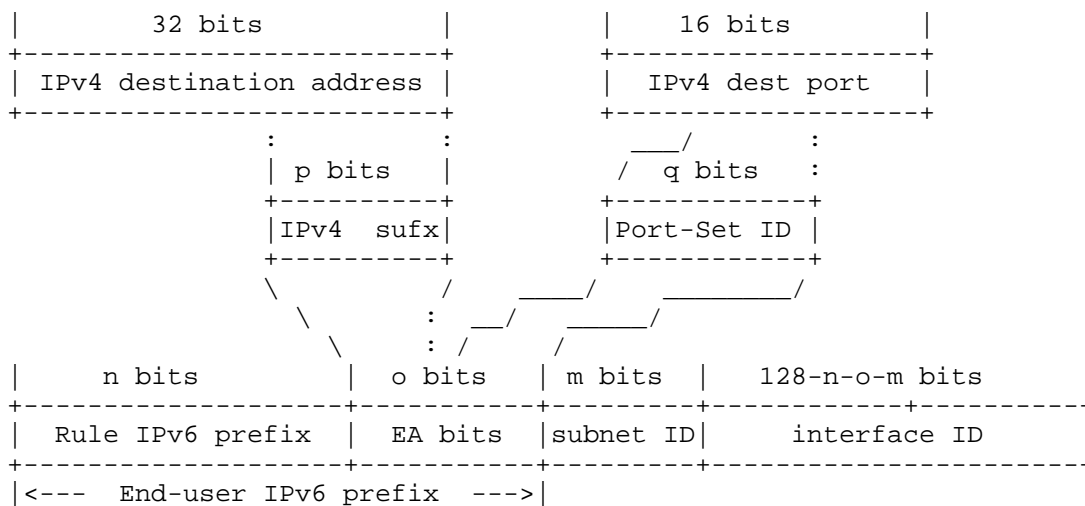


Figure 2: Deriving of MAP IPv6 address

### 3.1.2. 4rd-U

In 4rd-U [I-D.ietf-softwire-4rd-02], PSID itself is sufficient for defining a port set, as shown in Figure 3.

To derive the PSID from a given port, it only needs to take out the PSID bits from the 16bit port number.

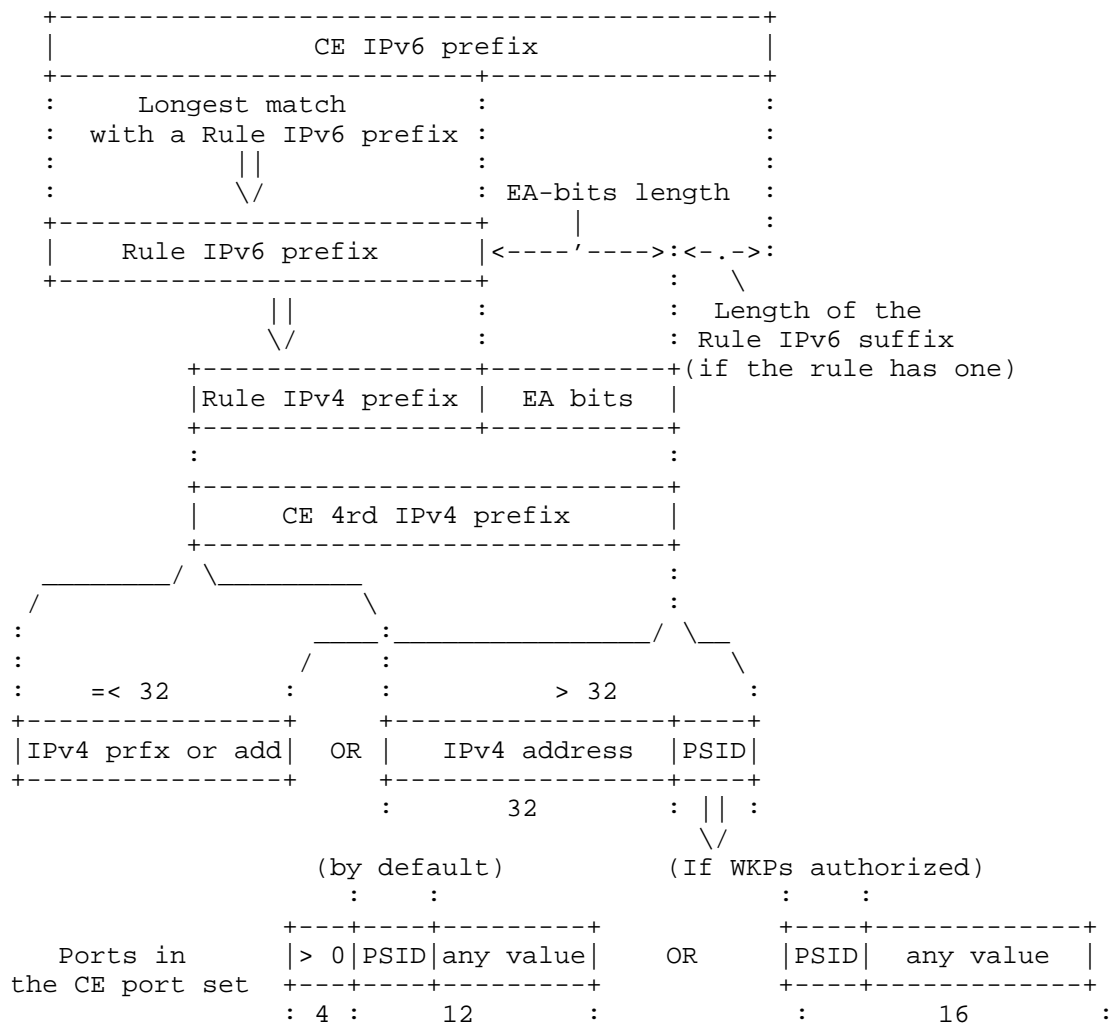


Figure 3: From CE IPv6 prefix to 4rd IPv4 address and Port set

## 3.1.3. Summary

Port set type	no-continuous
Stateless	yes
Security	good
Implementation	easy
Friendliness for NAT44	yes
Sharing ratio	up to $2^{12}$
Revert calculation from port number to PSID at BR	yes
Exclude well known ports	yes, 0~1023 or 0~4095

1. 4rd-U is a parameter-free algorithm, which is different MAP; while MAP can provide more variation due to the extra parameter(s). From the port set definition point of view, MAP and 4rd-U provide the same level of security.

2. MAP support sharing ratio up to  $2^{16}$ , although it may not be necessary.

## 3.2. Mask/Value style algorithms

[RFC6431] defines an IPCP option to allocate port set to CPEs, as shown in Figure 4.

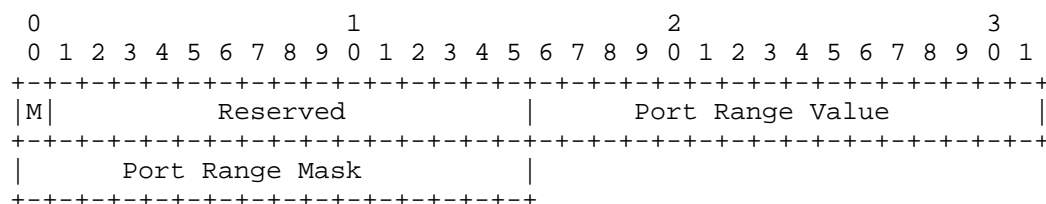


Figure 4: IPCP option format

The Port Range Value can be encoded in IPv6 address, similar as parameter PSID in other technologies, e.g. MAP [I-D.ietf-softwire-map-01].

To derive the Port Range Value from a given port, the port number should perform bit-and operation with the Port Range Mask.

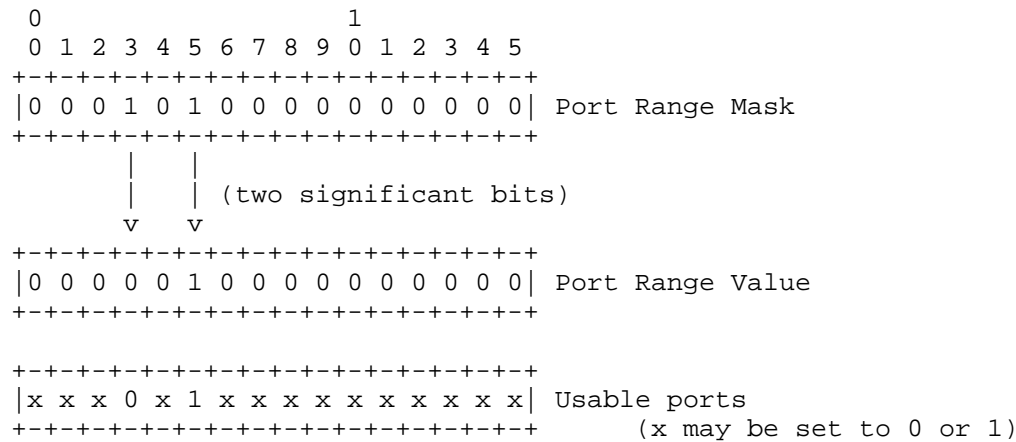


Figure 5: Example of Port Range Mask and Port Range Value

This alogrithm can have some kind of randomization effect by setting different number of bits and bits at different location in the Port Range Mask.

This algorithm may have a problem if the well known ports(0~1023 or 0~4096) need to be excluded, it is a bit difficult to achieve that. But if the operator do not have a specific usage for the well known ports, then it is OK to allocate those port to end users, just like other common ports. Some tests have done and prove that is OK.

Port set type	continuous, no-continuous
Stateless	yes
Security	good
Implementation	easy
Friendliness for NAT44	yes
Sharing ratio	up to 2^16
Revert calculation from port number to PSID at BR	yes
Exclude well known ports	difficult

### 3.3. Cryptographical style algorithms

The cryptographical port set definition algorithm introduced in [RFC6431] can provide very good security, but it is very difficult to derive the port set information, e.g. the starting point, from a given port. This algorithm can only be used in stateful scenarios, the BR must be operated in stateful mode.

In order to use this kind of algorithm in a stateless scenario, the algorithm must be reversible, that is, with some given information, it should be able to derive the port set information from a given port number.

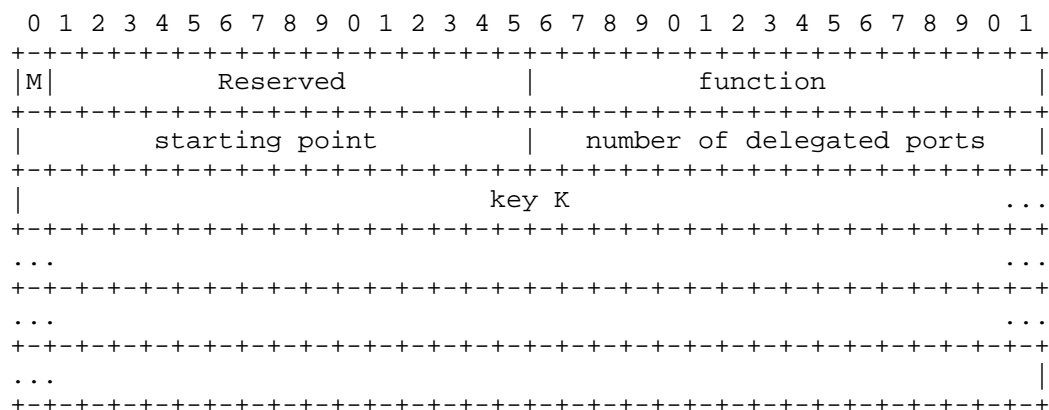


Figure 6: Format of the Cryptographically Random Port Range Option

Port set type	continuous, no-continuous
Stateless	No *
Security	Very good
Implementation	difficult
Friendliness for NAT44	yes
Sharing ratio	up to $2^{16}$
Revert calculation from port number to PSID at BR	No *
Exclude well known ports	difficult

\* It may be possible to find a cryptographic algorithm which can be reversed, e.g. define a reversible one-to-one mapping algorithm. But that is out the scope of this memo. If strong security is required, it may be worth giving this topic further study.

#### 4. Conclusion

TBD.

#### 5. IANA Considerations

This memo includes no request to IANA.

#### 6. Security Considerations

The port set should be as random as possible, in order to make it difficult to predict what the next port will be used, to avoid some potential TCP attack [RFC6056].

#### 7. References

##### 7.1. Normative References

[I-D.ietf-software-4rd-02]  
Despres, R., Penno, R., Lee, Y., Chen, G., and S. Jiang,

"IPv4 Residual Deployment via IPv6 - a unified Stateless Solution (4rd) (Work in progress)", Jan 2012.

[I-D.ietf-softwire-map-01]

Troan, O., Dec, W., Li, X., Bao, C., Zhai, Y., Matsushima, S., and T. Murakami, "Mapping of Address and Port (MAP) (Work in progress)", Jun 2012.

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

## 7.2. Informative References

[I-D.ietf-softwire-stateless-port-index-analysis]

Boucadair, M., Skoberne, N., and W. Dec, "Analysis of Port Indexing Algorithms", Sept 2011.

## Authors' Addresses

Tina Tsou (editor)  
Huawei Technologies (USA)  
2330 Central Expressway  
Santa Clara CA 95050  
USA

Phone: +1 408 330 4424  
Email: tina.tsou.zouting@huawei.com

Tetsuya Murakami  
IP Infusion  
1188 East Arques Avenue  
Sunnyvale  
USA

Email: tetsuya@ipinfusion.com

Simon Perreault  
Viagenie  
246 Aberdeen  
Quebec, QC G1R 2E1  
Canada

Phone: +1 418 656 9254  
Email: [simon.perreault@viagenie.ca](mailto:simon.perreault@viagenie.ca)  
URI: <http://viagenie.ca>





Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: January 5, 2013

J. Schoenwaelder  
Jacobs University  
C. Zhou  
Huawei Technologies  
T. Tsou  
Huawei Technologies (USA)  
C. Xie  
China Telecom  
July 4, 2012

Definition of Managed Objects for Lightweight 4over6 Transition  
Technology  
draft-zhou-softwire-lw4o6-mib-00

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects for managing Lightweight 4over6 transition technology.

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. The Internet-Standard Management Framework . . . . .	3
3. Conventions . . . . .	3
4. Overview . . . . .	3
5. Relationship to Other MIB Modules . . . . .	4
6. Definitions . . . . .	4
7. Security Considerations . . . . .	12
8. IANA Considerations . . . . .	13
9. Acknowledgements . . . . .	13
10. References . . . . .	13
10.1. Normative References . . . . .	13
10.2. Informative References . . . . .	14

## 1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols. In particular, it defines objects for managing the Lightweight 4over6 transition technology. The management of the network address translation function of Lightweight 4over6 initiators is expected to be handled by an updated version of the NAT-MIB [RFC4008], perhaps with a small Lightweight 4over6 specific addition.

## 2. The Internet-Standard Management Framework

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

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

## 3. 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].

## 4. Overview

The MIB module is organized into a group of scalars and tables.  
TODO: Add more details!

The OID tree structure of the MIB module is shown below.

```

--lw4o6Mib(1.3.6.1.2.1.XXXX)
+--lw4o6Objects(1)
+--lw4o6IfTable(1)
|   +--lw4o6IfEntry(1) [lw4o6IfIndex]
|   |   +-- --- InterfaceIndex lw4o6IfIndex(1)
|   |   +-- rwn Bits           lw4o6IfIcmpControl(2)
|   |   +-- r-n Counter32      lw4o6IfBindingMatchFailures(3)
|   |   +-- r-n Counter32      lw4o6IfNoBindingFailures(4)
+--lw4o6BindTable(2)
+--lw4o6BindEntry(1) [lw4o6BindIfIndex,lw4o6BindIndex]
+-- --- InterfaceIndex lw4o6BindIfIndex(1)
+-- --- BindingIndex   lw4o6BindIndex(2)
+-- r-n Enumeration    lw4o6BindType(3)
+-- r-n InetAddressIPv6 lw4o6BindIPv6Address(4)
+-- r-n InetAddressIPv4 lw4o6BindIPv4Address(5)
+-- r-n Integer32       lw4o6BindPortRangeValue(6)
+-- r-n Integer32       lw4o6BindPortRangeMask(7)
+-- r-n Integer32       lw4o6BindRandomFunction(8)
+-- r-n Integer32       lw4o6BindRandomStartingPoint(9)
+-- r-n OctetString     lw4o6BindRandomKey(10)
+-- r-n Integer32       lw4o6BindNumberOfPorts(11)
+-- r-n Gauge32         lw4o6BindNumberOfPortsUsed(12)
+-- r-n Counter32       lw4o6BindPortAllocationFailures(13)

```

## 5. Relationship to Other MIB Modules

The MIB module IMPORTS definitions from SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], SNMP-FRAMEWORK-MIB [RFC3411], IF-MIB [RFC2863], and INET-ADDRESS-MIB [RFC4001].

## 6. Definitions

LW4O6-MIB DEFINITIONS ::= BEGIN

### IMPORTS

```

MODULE-IDENTITY, OBJECT-TYPE,
Integer32, Gauge32, Counter32, mib-2
    FROM SNMPv2-SMI
    TEXTUAL-CONVENTION
    FROM SNMPv2-TC
OBJECT-GROUP, MODULE-COMPLIANCE
    FROM SNMPv2-CONF
InterfaceIndex
    FROM IF-MIB
InetAddressIPv4, InetAddressIPv6
    FROM INET-ADDRESS-MIB;

```

lw4o6Mib MODULE-IDENTITY

LAST-UPDATED "201207040000Z"

ORGANIZATION

"Huawei Technologies"

CONTACT-INFO

"Cathy Zhou  
Huawei Technologies  
Email: cathyzhou@huawei.com

Tina Tsou  
Huawei Technologies (USA)  
Email: tina.tsou.zouting@huawei.com"

DESCRIPTION

"The MIB module for managing the Lightweight 4over6 transition technology.

Copyright (c) 2012 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>)."

REVISION "201207040000Z"

DESCRIPTION

"Initial version, published as RFC XXXX."

-- RFC Ed.: replace XXXX with actual RFC number & remove this note  
::= { mib-2 XXXX }

lw4o6Notifications OBJECT IDENTIFIER ::= { lw4o6Mib 0 }  
lw4o6Objects OBJECT IDENTIFIER ::= { lw4o6Mib 1 }  
lw4o6Conformance OBJECT IDENTIFIER ::= { lw4o6Mib 2 }

-- Textual convention definitions:

BindingIndex ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"A unique value, greater than zero, identifying a Lightweight 4over6 binding. The value for each binding must remain constant at least from one re-initialization of the Lightweight 4over6 subsystem to the next re-initialization."

SYNTAX Integer32 (1..2147483647)

-- Object definitions:

-- lw4o6IfTable:

```
lw4o6IfTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Lw4o6IfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The lw4o6IfTable extends the interface table providing
        information about Lightweight 4over6 specific error conditions
        and it controls the interface specific handling of detected
        error situations."
    ::= { lw4o6Objects 1 }
```

```
lw4o6IfEntry OBJECT-TYPE
    SYNTAX      Lw4o6IfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry of the lw4o6IfTable providing information about
        Lightweight 4over6 statistics if an interface."
    INDEX { lw4o6IfIndex }
    ::= { lw4o6IfTable 1 }
```

```
Lw4o6IfEntry ::= SEQUENCE {
    lw4o6IfIndex          InterfaceIndex,
    lw4o6IfIcmpControl    BITS,
    lw4o6IfBindingMatchFailures Counter32,
    lw4o6IfNoBindingFailures Counter32
}
```

```
lw4o6IfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The network interface the Lightweight 4over6 statistics are
        associated with."
    ::= { lw4o6IfEntry 1 }
```

```
lw4o6IfIcmpControl OBJECT-TYPE
    SYNTAX      BITS { icmpOnMatchFailure(0), icmpOnBindingFailure(1) }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This object controls the generation of ICMP messages on certain
        failures."
```

If the icmpOnMatchFailure(0) bit is set, then an ICMP message is generated when an encapsulated packet is received that does not match a valid binding.

If the icmpOnBindingFailure(1) bit is set, then an ICMP message is generated if a packet is received for which there is no valid binding."

::= { lw4o6IfEntry 2 }

lw4o6IfBindingMatchFailures OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of packets received over this interfaces that have been dropped by the concentrator because the IPv6 source address of the outer header or the IPv4 source address or the port number of the inner header did not match a valid binding."

::= { lw4o6IfEntry 3 }

lw4o6IfNoBindingFailures OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of packets received over this interfaces that have been dropped by the concentrator because the IPv4 address and port number of a received IPv4 packet does not match a valid binding."

::= { lw4o6IfEntry 4 }

-- lw4o6IfBindTable:

lw4o6BindTable OBJECT-TYPE

SYNTAX SEQUENCE OF Lw4o6BindEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The lw4o6IfBindTable extends the interface table providing information about Lightweight 4over6 bindings."

::= { lw4o6Objects 2 }

lw4o6BindEntry OBJECT-TYPE

SYNTAX Lw4o6BindEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry of the lw4o6IfBindTable providing information about



```

    Lightweight 4over6 bindings."
    INDEX { lw4o6BindIfIndex, lw4o6BindIndex }
    ::= { lw4o6BindTable 1 }

-- DISCUSS: Is the binding table a per interface table or a global
--          table?

Lw4o6BindEntry ::= SEQUENCE {
    lw4o6BindIfIndex      InterfaceIndex,
    lw4o6BindIndex        BindingIndex,
    lw4o6BindType         INTEGER,
    lw4o6BindIPv6Address  InetAddressIPv6,
    lw4o6BindIPv4Address  InetAddressIPv4,
    lw4o6BindPortRangeValue Integer32,
    lw4o6BindPortRangeMask Integer32,
    lw4o6BindRandomFunction Integer32,
    lw4o6BindRandomStartingPoint Integer32,
    lw4o6BindRandomKey    OCTET STRING,
    lw4o6BindNumberOfPorts Integer32,
    lw4o6BindNumberOfPortsUsed Gauge32,
    lw4o6BindPortAllocationFailures Counter32
}

lw4o6BindIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The interface the Lightweight 4over6 bindings are
        associated with."
    ::= { lw4o6BindEntry 1 }

lw4o6BindIndex OBJECT-TYPE
    SYNTAX      BindingIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index uniquely identifying a binding."
    ::= { lw4o6BindEntry 2 }

lw4o6BindType OBJECT-TYPE
    SYNTAX      INTEGER { unknown(0), portrange(1), portrandom(2) }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The port binding type:

        portrange(1)  The port range is specified using a port

```

range value and a port range mask.

```
    portrandom(2)

    "
 ::= { lw4o6BindEntry 3 }

lw4o6BindIPv6Address OBJECT-TYPE
    SYNTAX      InetAddressIPv6
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The IPv6 address used with this binding."
    ::= { lw4o6BindEntry 4 }

lw4o6BindIPv4Address OBJECT-TYPE
    SYNTAX      InetAddressIPv4
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The IPv4 address used with this binding."
    ::= { lw4o6BindEntry 5 }

lw4o6BindPortRangeValue OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The port value used with this binding if the binding type
         is portrange."
    REFERENCE
        "draft-bajko-pripaddrassign-04"
    ::= { lw4o6BindEntry 6 }

lw4o6BindPortRangeMask OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The port mask used with this binding if the binding type
         is portrange."
    REFERENCE
        "draft-bajko-pripaddrassign-04"
    ::= { lw4o6BindEntry 7 }

lw4o6BindRandomFunction OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  read-only
```

STATUS current  
DESCRIPTION  
"The random function used with this binding if the binding type  
is portrandom."  
REFERENCE  
"draft-bajko-pripaddrassign-04"  
::= { lw4o6BindEntry 8 }

lw4o6BindRandomStartingPoint OBJECT-TYPE  
SYNTAX Integer32 (0..65535)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The value used as input to the random function if the  
binding type is portrandom."  
REFERENCE  
"draft-bajko-pripaddrassign-04"  
::= { lw4o6BindEntry 9 }

lw4o6BindRandomKey OBJECT-TYPE  
SYNTAX OCTET STRING  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The key used as input to the random function if the  
binding type is portrandom."  
REFERENCE  
"draft-bajko-pripaddrassign-04"  
::= { lw4o6BindEntry 10 }

lw4o6BindNumberOfPorts OBJECT-TYPE  
SYNTAX Integer32 (0..65535)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The total number of ports of this binding."  
::= { lw4o6BindEntry 11 }

lw4o6BindNumberOfPortsUsed OBJECT-TYPE  
SYNTAX Gauge32 (0..65535)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"The number of ports of this binding that are currently used."  
::= { lw4o6BindEntry 12 }

lw4o6BindPortAllocationFailures OBJECT-TYPE  
SYNTAX Counter32

```
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The number of situations where a port could not be allocated
    because there we no more ports left in the binding."
 ::= { lw4o6BindEntry 13 }

-- Compliance definitions:

lw4o6Groups      OBJECT IDENTIFIER ::= { lw4o6Conformance 1 }
lw4o6Compliances OBJECT IDENTIFIER ::= { lw4o6Conformance 2 }

lw4o6FullCompliance MODULE-COMPLIANCE
    STATUS        current
    DESCRIPTION
        "Compliance statement for implementations supporting
        read/write access, according to the object definitions."
    MODULE         -- this module

    MANDATORY-GROUPS {
        lw4o6IfGroup,
        lw4o6BindGroup
    }

    ::= { lw4o6Compliances 1 }

lw4o6ReadOnlyCompliance MODULE-COMPLIANCE
    STATUS        current
    DESCRIPTION
        "Compliance statement for implementations supporting
        only readonly access."
    MODULE         -- this module

    MANDATORY-GROUPS {
        lw4o6IfGroup,
        lw4o6BindGroup
    }

    OBJECT lw4o6IfIcmpControl
    MIN-ACCESS    read-only
    DESCRIPTION
        "Write access is not required."

    ::= { lw4o6Compliances 2 }

lw4o6IfGroup OBJECT-GROUP
    OBJECTS {
```

```
        -- lw4o6IfIndex,
        lw4o6IfIcmpControl,
        lw4o6IfBindingMatchFailures,
        lw4o6IfNoBindingFailures
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        performance of a Lightweight 4over6 interface."
    ::= { lw4o6Groups 1 }

lw4o6BindGroup OBJECT-GROUP
    OBJECTS {
        -- lw4o6BindIfIndex,
        -- lw4o6BindIndex
        lw4o6BindType,
        lw4o6BindIPv6Address,
        lw4o6BindIPv4Address,
        lw4o6BindPortRangeValue,
        lw4o6BindPortRangeMask,
        lw4o6BindRandomFunction,
        lw4o6BindRandomStartingPoint,
        lw4o6BindRandomKey,
        lw4o6BindNumberOfPorts,
        lw4o6BindNumberOfPortsUsed,
        lw4o6BindPortAllocationFailures
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing insight into the
        bindings associated with a Lightweight 4over6 interface."
    ::= { lw4o6Groups 2 }

END
```

## 7. Security Considerations

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

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to

control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

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.

## 8. IANA Considerations

IANA is requested to assign a value for "XXXX" under the 'mib-2' subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "XXXX" (here and in the MIB module) with the assigned value and to remove this note.

## 9. Acknowledgements

TBD

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J.

Schoenwaelder, Ed., "Textual Conventions for SMIV2",  
STD 58, RFC 2579, April 1999.

- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder,  
"Conformance Statements for SMIV2", STD 58, RFC 2580,  
April 1999.
- [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", STD 62, 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  
C. Wang, "Definitions of Managed Objects for Network  
Address Translators (NAT)", RFC 4008, March 2005.

#### 10.2. Informative References

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

#### Authors' Addresses

Juergen Schoenwaelder  
Jacobs University  
Campus Ring 1  
Bremen 28759  
Germany

EMail: j.schoenwaelder@jacobs-university.de

Cathy Zhou  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen 518129  
P.R. China

EMail: cathy.zhou@huawei.com

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

EMail: tina.tsou.zouting@huawei.com

Chongfeng Xie  
China Telecom  
Room 708 No.118, Xizhimenneidajie  
Beijing  
P.R. China

EMail: xiechf@ctbri.com.cn



