Softwire WG
Internet-Draft
Intended status: Standards Track
Expires: July 22, 2013

M. Boucadair
France Telecom
I. Farrer
Deutsche Telekom

January 18, 2013

Unified IPv4-in-IPv6 Softwire CPE
draft-bfmk-softwire-unified-cpe-02

Abstract

Transporting IPv4 packets encapsulated in IPv6 is a common solution to the problem of IPv4 service continuity over IPv6-only provider networks. A number of differing functional approaches have been developed for this, each having their own specific characteristics. As these approaches share a similar functional architecture and use the same data plane mechanisms, this memo describes a specification whereby a single CPE can interwork with all of the standardized and proposed approaches to providing encapsulated IPv4 in IPv6 services.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

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

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

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

This Internet-Draft will expire on July 22, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
1.1. Rationale 3
2. IPv4 Service Continuity Architectures: A 'Big Picture' Overview 4
2.1. Functional Elements 5
2.2. Required Provisoning Information 6
3. Unified Softwire CPE Behaviour 7
3.1. IPv4 Address Functional Requirements 7
3.2. Generic CPE Bootstrapping Logic 7
3.3. Customer Side DHCP Based Provisioning 9
3.4. Forwarding Action by the Customer End-Node 11
4. Security Considerations 11
5. IANA Considerations 11
6. Acknowledgements 11
7. References 11
7.1. Normative References 11
7.2. Informative References 12
Authors' Addresses 12

1. Introduction

IPv4 service continuity is one of the major technical challenges which must be considered during IPv6 migration. Over the past few years, a number of different approaches have been developed to assist with this problem. These approaches, or modes, exist in order to meet the particular deployment, scaling, addressing and other requirements of different service provider's networks. Section 2 of this document describes these approaches in more detail.

A common feature shared between all of the differing modes is the integration of softwire tunnel end-point functionality into the CPE router. Due to this inherent data plane similarity, a single CPE may be capable of supporting several different approaches. Users may also wish to configure a specific mode of operation.

A service provider's network may also have more than one mode enabled in order to support diverse CPE client functionality, during migration between modes or where services require specific supporting softwire architectures.

For softwire based services to be successfully established, it is essential that the customer end-node, the service provider end-node and provisioning systems are able to indicate their capabilities and preferred mode of operation.

This memo describes the logic required by both the CPE tunnel end-node and the service provider's provisioning infrastructure so that softwire services can be provided in mixed-mode environments.

1.1. Rationale

The following rationale has been adopted for this document:

(1) Describe the functionality of each the different solution modes and provide clear distinctions between them
(2) Simplify solution migration paths: Define unified CPE behavior, allowing for smooth migration between the different modes
(3) Deterministic CPE co-existence behavior: Specify the behavior when several modes co-exist in the CPE
(4) Deterministic service provider co-existence behavior: Specify the behavior when several modes co-exist in the service providers network
(5) Re-usability: Maximize the re-use of existing functional blocks including tunnel end-points, port restricted NAPT44, forwarding behavior, etc.

(6) Solution agnostic: Adopt neutral terminology and avoid (as far as possible) overloading the document with solution-specific terms
(7) Flexibility: Allow operators to compile CPE software only for the mode(s) necessary for their chosen deployment context(s)
(8) Simplicity: Provide a model that allows operators to only implement the specific mode(s) that they require without the additional complexity of unneeded modes.

2. IPv4 Service Continuity Architectures: A 'Big Picture' Overview

The solutions which have been proposed within the Softwire WG can be categorized into three main functional approaches, differentiated by the amount and type of state that the service provider needs to maintain within their network:

(1) Full stateful approach (DS-Lite, [RFC6333]): Requires per-session state to be maintained in the Service Provider's network.
(2) Binding approach (e.g., Lightweight 4over6 (Lw4o6) [I-D.cui-softwire-b4-translated-ds-lite][I-D.ietf-softwire-public-4over6] or MAP 1:1 [I-D.ietf-softwire-map]): Requires a single per-subscriber state (or a few) to be maintained in the Service Provider's network.
(3) Full stateless approach (MAP, [I-D.ietf-softwire-map]): Does not require per-session or per-subscriber state to be maintained in the Service Provider's network.

All these approaches share a similar architecture, with a tunnel endpoint located in the CPE and a remote tunnel endpoint. All use IPv6 as the transport protocol for the delivery of an IPv4 connectivity service using an IPv4-in-IPv6 encapsulation scheme [RFC2473].

Several cases can be envisaged:

1. The CPE is complied to support only one mode: No issue is raised by this case.
2. The CPE supports several modes but only one mode is explicitly configured: No issue is raised by this case.
3. The CPE supports several modes but no mode is explicitly enabled: the CPE will need additional triggers to decide which mode to activate.
4. The CPE supports several modes and several modes are configured: the CPE will need additional triggers to decide which mode to activate.

As this document describes a provisioning profile whereby a single CPE could be capable of supporting any, or multiple modes, the

customer should not be required to have any knowledge of the capabilities and configuration of their CPE, or of their service provider's network.

The service provider, however, may have only a single mode enabled, or may have multiple modes, but with one preferred mode. For this reason, it is necessary to approach the configuration of CPEs from the standpoint of the service provider's network capabilities.

2.1. Functional Elements

The functional elements for each of the solution modes are listed in Table 1:

Mode	Customer side	Network side
DS-Lite	B4	AFTR
Lw4o6	lwB4	lwAFTR
MAP	MAP CE	MAP BR

Table 1: Functional Elements

Table 2 describes each functional element:

Functional Element	Description
B4	An IPv4-in-IPv6 tunnel endpoint; the B4 creates a tunnel to a pre-configured remote tunnel endpoint.
AFTR	Provides both an IPv4-in-IPv6 tunnel endpoint and a NAT44 function implemented in the same node.
lwB4	A B4 which supports port-restricted IPv4 addresses. An lwB4 MAY also provide a NAT44 function.
lwAFTR	An IPv4-in-IPv6 tunnel endpoint which maintains per-subscriber address binding. Unlike the AFTR, it MUST NOT perform a NAPT44 function.
MAP CE	A B4 which supports port-restricted IPv4 addresses. It MAY be co-located with a NAT44. A MAP CE forwards IPv4-in-IPv6 packets using provisioned mapping rules to derive the remote tunnel endpoint.
MAP BR	An IPv4-in-IPv6 tunnel endpoint. A MAP BR forwards IPv4-in-IPv6 packets following pre-configured mapping rules.

Table 2: Required Element Functionality

Table 3 identifies features required by the customer end-node.

Functional Element	IPv4-in-IPv6 tunnel endpoint	Port-restricted IPv4	Port-restricted NAT44
B4	Yes	N/A	No
lwB4	Yes	Yes	Optional
MAP-E CE	Yes	Yes	Optional

Table 3: Supported Features

2.2. Required Provisoning Information

Table 4 identifies the provisioning information required for each solution mode.

Mode	Provisioning Information
DS-Lite	Remote IPv4-in-IPv6 Tunnel Endpoint Address
Lw4o6	Remote IPv4-in-IPv6 Tunnel Endpoint Address
	IPv4 Address
	Port Set
MAP-E	Mapping Rules
	MAP Domain Parameters

Table 4: Provisioning Information

Note: MAP Mapping Rules are translated into the following configuration parameters: Set of remote IPv4-in-IPv6 tunnel endpoint addresses, IPv4 address and port set.

Note: Required provisioning information for each mode may also be represented as follows:

```
DS-Lite: - Remote IPv4-in-IPv6 Tunnel Endpoint
  Lw4o6: - DS-Lite set of provisioning information
         - IPv4 address
         - Port set
  MAP-E: - Lw4o6 set of provisioning information
```

- Forwarding mapping rules

3. Unified Softwire CPE Behaviour

This section specifies a unified CPE behavior capable of supporting any one, or combination of, the three modes.

3.1. IPv4 Address Functional Requirements

The following two requirements must be met by the functional elements:

Full IPv4 Address Assingment All the aforementioned modes MUST be designed to allow either a full or a shared IPv4 address to be assigned to a customer end-node. DS-Lite and MAP-E fulfill this requirement. With minor changes, the [I-D.cui-softwire-b4-translated-ds-lite] specification can be updated to assign full IPv4 addresses.

Customer End-Node NAT A NAT function within the customer end-node is not required for DS-Lite, while it is optional for both MAP-E and Lw4o6. When NAT is enabled for MAP-E or Lw4o6, the customer end-node NAT MUST be able to restrict the external translated source ports to the set of ports that it has been provisioned with.

3.2. Generic CPE Bootstrapping Logic

The generic provisioning logic is designed to meet the following requirements:

- When several service continuity modes are supported by the same CPE, it MUST be possible to configure a single mode for use.
- For each network attachment, the end-node MUST NOT activate more than one mode.
- The CPE MAY be configured by a user or via remote device management means (e.g., DHCP, TR-069).
- A network which supports one or several modes MUST return valid configuration data enabling requesting devices to unambiguously select a single mode to use for attachment.
- A CPE which supports only one mode or it is configured to enable only mode MUST ignore any configuration parameter which is not required for the mode it supports.

This section sketches a generic algorithm to be followed by a CPE supporting one or more of the modes listed above. Based on the retrieved information, the CPE will determine which mode to activate.

(1) If a given mode is enabled (DS-Lite, Lw4o6 or MAP-E), the CPE MUST be configured with the required provisioning information listed in Table 4. If all of the required information is not available locally, the CPE MUST use available provisioning means (e.g., DHCP) to retrieve the missing configuration data.

(2) If the CPE supports several modes, but no mode is explicitly enabled, the CPE MUST use available provisioning means (e.g., DHCP) to retrieve available configuration parameters and use the availability of individual parameters to ascertain which functional mode to configure:

(2.1) If only a Remote IPv4-in-IPv6 Tunnel Endpoint is received, the CPE MUST proceed as follows:
(2.1.1) IPv4-in-IPv6 tunnel endpoint initialization is defined in [RFC6333].
(2.1.2) Outbound IPv4 packets are forwarded to the next hop as specified in Section 3.4.

(2.2) If a Remote IPv4-in-IPv6 Tunnel Endpoint, an IPv4 Address and optionally a Port Set are received, the CPE MUST behave as follows:
(2.2.1) IPv4-in-IPv6 tunnel endpoint initialization is similar to the B4 [RFC6333].
(2.2.2) When NAPT44 is required (e.g., because the CPE is a router), a NAPT44 module is enabled.
(2.2.3) The tunnel endpoint address is selected from the native IPv6 addresses configured on the CPE. No particular considerations are required to be taken into account to generate the Interface Identifier.
(2.2.4) When a port set is provisioned, the external source ports MUST be restricted to the provisioned set of ports.
(2.2.5) After translation, outbound IPv4 packets are forwarded to the next hop as specified in Section 3.4.

(2.3) If Mapping Rule(s) are received, the CPE MUST behave as follows:

(2.3.1) IPv4-in-IPv6 tunnel endpoint initialization is similar to the B4 [RFC6333].
(2.3.2) The tunnel endpoint is assigned with an IPv6 address which includes an IPv4 address. The MAP Interface Identifier is based on the format specified in Section 2.2 of [RFC6052].
(2.3.3) When NAPT44 is required (e.g., because the CPE is a router), a NAPT44 module is enabled.
(2.3.4) When a port set is provisioned, the external source port MUST be restricted to the provisioned set of ports.
(2.3.5) After translation, outbound IPv4 packets then forwarded to the next hop as specified in Section 3.4.

3.3. Customer Side DHCP Based Provisioning

[DISCUSSION NOTE:

1. This section will be updated to reflect the consensus from DHC WG.

2. As it is proposed that OPTION_MAP would be used for all new softwire provisioning, should we rename OPTION_MAP to OPTION_SW (incl. the associated sub-options)?]

]

DHCP-based configuration SHOULD be implemented by the customer end-node using the following two DHCP options:

OPTION_AFTR_NAME [RFC6334] Provides the FQDN for the remote IPv4-in-IPv6 tunnel end-point.

OPTION_MAP [I-D.ietf-softwire-map-dhcp] Provides IPv4-related configuration for binding mode and/or mapping rules for stateless mode (including MAP parameters such as offset, domain prefix, etc.). OPTION_MAP_BIND is a sub-option used to convey an IPv4 address (for example, encoded as an IPv4-mapped IPv6 address [RFC4291]). This address is used when binding mode is enabled. The receipt of OPTION_MAP_BIND is an implicit indication to the customer side device to operate in binding, rather than stateless mode.

The customer end-node uses the DHCP Option Request Option (ORO) to request either one or both of these options depending on which modes

it is capable of and configured to support.

The DHCP option(s) sent in the response allow the service provider to inform the customer end-node which operating mode to enable.

The following table shows the different DHCP options (and sub-options) that the service provider can supply in a response.

DHCP Option	Stateful Mode	Binding Mode	Stateless Mode
OPTION_AFTR_NAME	Yes	Yes	Optional
OPTION_MAP_BIND	No	Yes	No
OPTION_MAP_RULE	No	No	Yes
OPTION_MAP_PORTPARAMS	No	Optional	Optional

Table 5: DHCP Option Provisioning Matrix

The customer side device MUST interpret the received DHCP configuration parameters according to the logic defined in Section 3.2:

- If only OPTION_AFTR_NAME is received, then the device MUST operate in stateful mode
- If both OPTION_AFTR_NAME and OPTION_MAP_BIND are received then the device MUST operate in binding mode
- If one or more OPTION_MAP_RULE options are received, then the customer side device MUST operate in stateless mode
- If both OPTION_AFTR_NAME and OPTION_MAP_RULE(s) are received, then the customer side device MUST operate as a MAP CE. OPTION_AFTR_NAME provides the FQDN of the MAP BR.
- If OPTION_MAP_PORTPARAMS is received as a sub-option to either OPTION_MAP_BIND or OPTION_MAP_RULE, then NAPT44 MUST be configured using the supplied port-set for external translated source ports.

From the service providers side, the following rule MUST be followed:

- The DHCP server MUST NOT send both OPTION_MAP_BIND and OPTION_MAP_RULE in a single OPTION_MAP response.

3.4. Forwarding Action by the Customer End-Node

For all modes, the longest prefix match algorithm MUST be enforced to forward outbound IPv4 packets.

Specifically, this algorithm will:

- Always return the address of the AFTR for the DS-Lite mode.
- Always return the address of the lwAFTR for the binding mode.
- Return the next hop according to the pre-configured mapping rules for the stateless mode (i.e., MAP-E).

4. Security Considerations

Security considerations discussed in Section 7 of [I-D.ietf-softwire-stateless-4v6-motivation] and Section 11 of [RFC6333] should be taken into account.

5. IANA Considerations

This document does not require any action from IANA.

6. Acknowledgements

Many thanks to T. Tsou, S. Perrault, S. Sivakumar, O. Troan, W. Dec, M. Chen, for their review and comments.

Special thanks to S. Krishnan for the suggestions and guidance.

7. References

7.1. Normative References

[I-D.cui-softwire-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-softwire-b4-translated-ds-lite-09 (work in progress), October 2012.

[I-D.ietf-softwire-map]
Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., and T. Murakami, "Mapping of Address and Port with

Encapsulation (MAP)", draft-ietf-softwire-map-02 (work in progress), September 2012.

[I-D.ietf-softwire-map-dhcp]
Mrugalski, T., Troan, O., Bao, C., Dec, W., and L. Yeh, "DHCPv6 Options for Mapping of Address and Port", draft-ietf-softwire-map-dhcp-01 (work in progress), August 2012.

[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.

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

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

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

7.2. Informative References

[I-D.ietf-softwire-public-4over6]
Cui, Y., Wu, J., Wu, P., Vautrin, O., and Y. Lee, "Public IPv4 over IPv6 Access Network", draft-ietf-softwire-public-4over6-04 (work in progress), October 2012.

[I-D.ietf-softwire-stateless-4v6-motivation]
Boucadair, M., Matsushima, S., Lee, Y., Bonness, O., Borges, I., and G. Chen, "Motivations for Carrier-side Stateless IPv4 over IPv6 Migration Solutions", draft-ietf-softwire-stateless-4v6-motivation-05 (work in progress), November 2012.

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

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes
France

Email: mohamed.boucadair@orange.com

Ian Farrer
Deutsche Telekom
Germany

Email: ian.farrer@telekom.de

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 01, 2014

Y. Chen
J. Wu
Tsinghua University
X. Tang
G. Zhou
China Unicom Research Institute
August 28, 2013

Gateway-Initiated 4over6 Deployment
draft-chen-softwire-gw-init-4over6-02

Abstract

Gateway-Initiated 4over6 is a variant of Lightweight 4over6. A Lightweight B4 in Lightweight 4over6 mechanism is a router which acts as a tunnel initiator for the IPv4-in-IPv6 tunnel. This mechanism mainly focuses on the scenario in which an IPv4 address and related configuration information is configured to the device behind Lightweight B4. Gateway-Initiated 4over6 uses the full IPv4 address rather than a shared address. This enables an unmodified end server or host that is behind a Lightweight B4 to get access to the IPv4 Internet through an IPv6 network.

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 March 01, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents

(http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. Terminology 2
3. Requirements Language 3
4. GI-4over6 Architecture 3
5. GI-4over6 in ICP Network 4
5.1. Static Configuration to Establish Tunnel 4
5.2. Dynamic Configuration to Establish Tunnel 5
6. 4over6 Gateway Data Plane Behaviors 5
7. Security Considerations 5
8. IANA Considerations 5
9. References 5
9.1. Normative References 5
9.2. Informative References 5
Authors' Addresses 6

1. Introduction

In typical use case of Lightweight 4over6 (Lw4over6) ([I-D.ietf-softwire-lw4over6]), IPv4 address (and available port set) is provisioned to the Lightweight B4 (LwB4), the tunnel initiator. However, there are some cases in which IPv4 address and related configuration are not be provisioned to LwB4, but the end device behind it. There is a typical scenario in this case, that is Lw4over6 is used in an Internet Content Provider (ICP) network, and the device behind LwB4 is an ICP server.

Gateway-Initiated 4over6 (GI-4over6) is a variant of Lw4over6. It mainly focuses on the scenario in which an IPv4 address and related configuration information is provisioned to the device behind LwB4. Provisioning full address is preferred to provisioning shared address (port-restricted address) in GI-4over6. It enables an unmodified IPv4 device that behind the LwB4 to get access to IPv4 Internet through IPv6 network.

2. Terminology

This document uses the terms defined in [I-D.ietf-softwire-lw4over6].

The other terms used are defined as follows:

- o End device: The device in the IPv4 network behind the 4over6 gateway. It can be an IPv4-only or a dual-stack device.
- o End server: The end device in an ICP network is supposed to be an end server.
- o 4over6 gateway: The dual-stack gateway device located at the border of both IPv4 and IPv6 networks. It should be configured with an IPv4 address and the IPv6 address of LwAFTR, and act as the LwB4 on the data plane.

3. 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].

4. GI-4over6 Architecture

The general architecture of GI-4over6 is illustrated as Figure 1. The 4over6 gateway is a dual-stack gateway device which establishes IPv4-in-IPv6 tunnel with the Lightweight Address Family Transition Router (LwAFTR) and performs the LwB4 function on data plane. The LwAFTR is a dual-stack border router deployed at the edge of the IPv6 network and the Internet. The IPv4 network can be either an ICP network, or a customer network of an ISP. The IPv6 network can be either an ICP access network or an ISP access network. Either or both of these networks could be dual-stack.

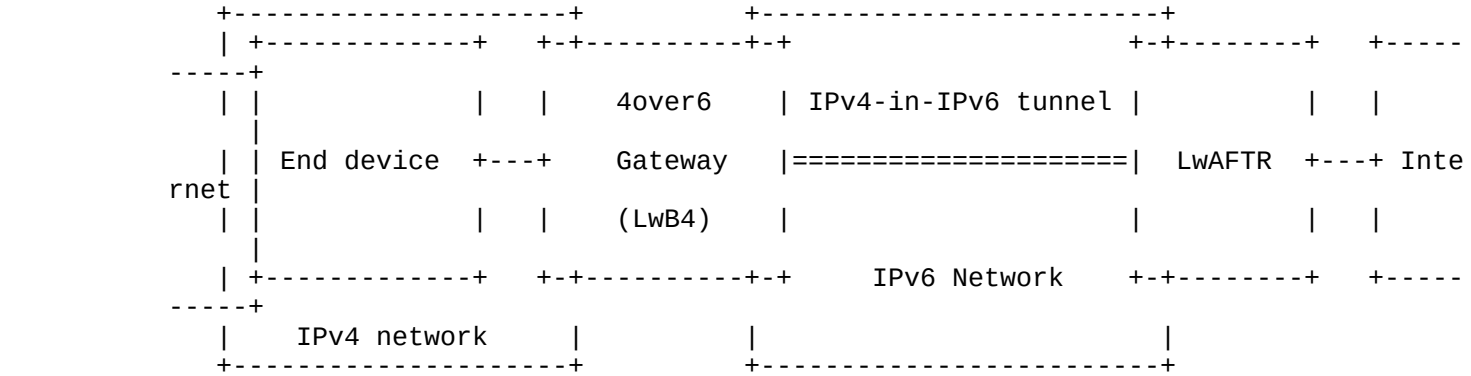


Figure 1 GI-4over6 Architecture

The 4over6 gateway is configured with an public IPv4 address on its

"left" side, an IPv6 address on its "right" side, either by static (in ICP network) or dynamic (in customer network) way. It is also configured with the IPv6 address of LwAFTR as the address of the tunnel endpoint. Each end device has a public IPv4 address with all ports (0-65535) available, hence there is no need to implement NAPT44 on the 4over6 gateway.

One typical scenario of this framework is that using Lw4over6 in an ICP network. There might be other similar scenarios, and they could be included in this document in the future.

5. GI-4over6 in ICP Network

Considering an ISP that plans to update its network to IPv6, one of the major issues it may be faced with is the update of its ICP network. If the ICP network is to be updated to run IPv6, the server in the network should also be updated to support IPv6. Obviously it is not trivial to update upper layer service running on the server to support a network layer protocol. It’s ideal if the ICP access network is updated to IPv6, but still capable of providing the server with access to IPv4 Internet, meanwhile the ICP network (and the servers inside) stay unmodified.

In this scenario, the end server has already been configured with a full public IPv4 address, and it’s expected to stay unchanged during the update of the network. It has also been configured with other IPv4 related configurations like the network mask of the IPv4 ICP network, the IPv4 address of DNS server, etc.

The 4over6 gateway has already been configured with the routing to the end server. It MUST establish the IPv4-in-IPv6 tunnel with the LwAFTR, in order to forward the IPv4 traffics between the end server and the IPv4 Internet. The establishment of the IPv4-in-IPv6 tunnel could be done either by static - the most likely way - or dynamic configuration.

5.1. Static Configuration to Establish Tunnel

The LwAFTR is statically configured with the binding of the public IPv4 address of the end server, the available port set (0-65535), and IPv6 address of the 4over6 gateway in its binding table statically.

In a more general case, the addresses of servers behind the same 4over6 gateway can aggregate. And as the 4over6 gateway and the LwAFTR are both managed by the ISP, people who configure the LwAFTR are usually aware of the routing to the ICP network behind the 4over6 gateway. Hence the LwAFTR can be configured with the following binding: the network prefix of the ICP network, the available port

set (0-65535), and IPv6 address of the 4over6 gateway.

5.2. Dynamic Configuration to Establish Tunnel

Dynamic configuration could be adopted in case the static configuration is not feasible or practical.

The 4over6 gateway MUST inform the LwAFTR of all of its IPv4 routing information (i.e. the whole IPv4 routing table). The detail of this process could be clarified in related draft in future.

Once the LwAFTR received the routing information from the 4over6 gateway, it should add the entry(s) into its binding table, with the given routing information. The binding may looks like: the ICP network prefix, available port set (0-65535), the IPv6 address of the 4over6 gateway.

6. 4over6 Gateway Data Plane Behaviors

The 4over6 gateway must perform the LwB4 function on the data plane. The data plane behavior of 4over6 gateway uses the description in section 5.2 of [I-D.ietf-softwire-lw4over6]. However, there is no need to implement NAPT44 function on 4over6 gateway, because each end server behind the 4over6 gateway has a public IPv4 address with all ports available.

7. Security Considerations

TBD

8. IANA Considerations

This document does not include an IANA request.

9. References

9.1. Normative References

[I-D.ietf-softwire-lw4over6]
Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture", draft-ietf-softwire-lw4over6-01 (work in progress), July 2013.

9.2. Informative References

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

Authors' Addresses

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

Phone: +86 10 6278 5822
Email: chenycmx@gmail.com

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

Xiongyan Tang
China Unicom Research Institute
33 Erlong Road, Xicheng District
Beijing 100032
P.R.China

Phone: +86 10 6652 2558
Email: tangxy@chinaunicom.cn

Guangtao Zhou
China Unicom Research Institute
9 Shouti South Road, Haidian District
Beijing 100048
P.R.China

Phone: +86 10 6789 9600
Email: zhouguangtao@chinaunicom.cn

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

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

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

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on August 19, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

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

1. Introduction

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

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

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

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

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

- NAPT44 functionality is already widely supported and used in today's CPE devices. Lw4o6 uses this to provide private<->public NAPT44, meaning that the service provider does not need a centralized NAT44 function.

- The amount of state that must be maintained centrally in the AFTR can be reduced from per-flow to per-subscriber. This reduces the amount of resources (memory and processing power) necessary in the AFTR.

- The reduction of maintained state results in a greatly reduced logging overhead on the service provider.

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

be deployed.

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

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

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

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

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

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Terminology

The document defines the following terms:

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

Lightweight B4 (lwB4):	A B4 element (Basic Bridging BroadBand element [RFC6333]), which supports Lightweight 4over6 extensions. An lwB4 is a function implemented on a dual-stack capable node, (either a directly connected device or a CPE), that supports port-restricted IPv4 address allocation, implements NAPT44 functionality and creates a tunnel to an lwAFTR
Lightweight AFTR (lwAFTR):	An AFTR element (Address Family Transition Router element [RFC6333]), which supports Lightweight 4over6 extension. An lwAFTR is an IPv4-in-IPv6 tunnel endpoint which maintains per-subscriber address binding only and does not perform a NAPT44 function.
Restricted Port-Set:	A non-overlapping range of allowed external ports allocated to the lwB4 to use for NAPT44. Source ports of IPv4 packets sent by the B4 must belong to the assigned port-set. The port set is used for all port aware IP protocols (TCP, UDP, SCTP etc.)
Port-restricted IPv4 Address:	A public IPv4 address with a restricted port-set. In Lightweight 4over6, multiple B4s may share the same IPv4 address, however, their port-sets must be non-overlapping.

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

4. Lightweight 4over6 Architecture

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

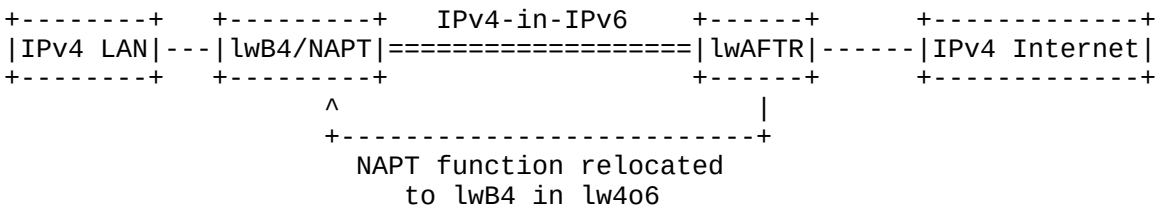


Figure 1 Lightweight 4over6 Data Plane Overview

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

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

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

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

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

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

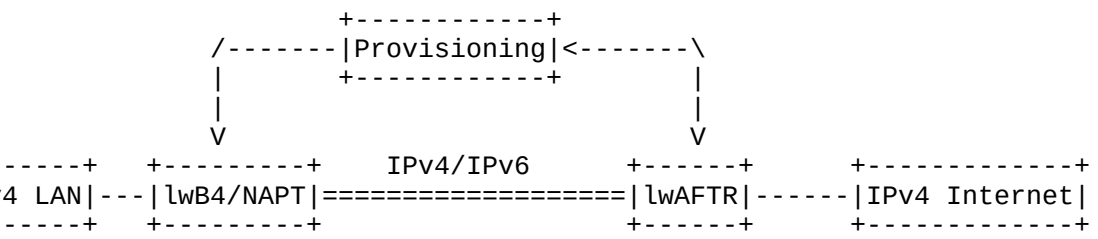


Figure 2 Lightweight 4over6 Provisioning Synchronization

5. Lightweight B4 Behavior

5.1. Lightweight B4 Provisioning

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

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

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

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

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

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

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

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

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

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

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

5.2. Lightweight B4 Data Plane Behavior

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

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

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

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

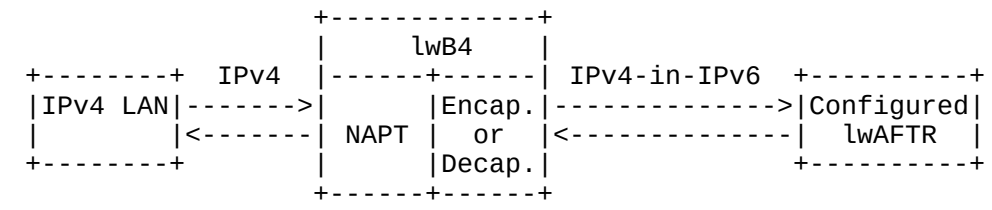


Figure 3 Working Flow of the lwB4

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

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

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

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

6. Lightweight AFTR Behavior

6.1. Binding Table Maintenance

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

- IPv6 Address for a single lwB4

- Public IPv4 Address
- Restricted port-set

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

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

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

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

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

6.2. lwAFTR Data Plane Behavior

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

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

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

the IPv4 destination.

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

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

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

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

7. Provisioning of IPv4 address and Port Set

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

- DHCP: Extending DHCP protocol MAY be used for the provisioning [I-D.ietf-dhc-dhcpv4-over-ipv6] [I-D.ietf-softwire-map-dhcp].

- PCP[I-D.ietf-pcp-base]: a lwB4 MAY use [I-D.tsou-pcp-natcoord] to retrieve a restricted IPv4 address and a set of ports.

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

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

8. ICMP Processing

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

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

Either:

1. Check the ICMP Type field.

2. If the ICMP type is set to 0 or 8 (echo reply or request), then the lwAFTR MUST take the value of the ICMP identifier field as the source port, and use this value to lookup the binding table for an encapsulation destination. If a match is found, the lwAFTR forwards the ICMP packet to the IPv6 address stored in the entry; otherwise it MUST discard the packet.

3. If the ICMP type field is set to any other value, then the lwAFTR MUST use the method described in REQ-3 of [RFC5508] to locate the source port within the transport layer header in ICMP packet's data field. The destination IPv4 address and source port extracted from the ICMP packet are then used to make a lookup in the binding table. If a match is found, it MUST forward the ICMP reply packet to the IPv6 address stored in the entry; otherwise it MUST discard the packet.

Or:

- Discard all inbound ICMP messages.

The ICMP policy SHOULD be configurable.

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

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

9. Security Considerations

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

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

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

10. IANA Considerations

This document does not include an IANA request.

11. Author List

The following are extended authors who contributed to the effort:

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

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

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

Phone: +86-10-62785822
Email: pengwu.thu@gmail.com

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

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

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

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

Xiaohong Deng
France Telecom

Email: xiaohong.deng@orange.com

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

Email: cathyzhou@huawei.com

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

Email: adurand@juniper.net

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

Email: repenno@cisco.com

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

Email: axel.clauberg@telekom.de

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

Email: lhoffman@bouyguestelecom.fr

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

Email: fibrib@gmail.com

12. Acknowledgement

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

This document is a merge of three documents: [I-D.cui-softwire-b4-translated-ds-lite], [I-D.zhou-softwire-b4-nat] and [I-D.penno-softwire-sdnat].

13. References

13.1. Normative References

[I-D.bfmk-softwire-unified-cpe]
Boucadair, M. and I. Farrer, "Unified IPv4-in-IPv6 Softwire CPE", draft-bfmk-softwire-unified-cpe-02 (work in progress), January 2013.

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

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

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

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

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

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

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

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-

Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

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

[RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.

[RFC6431] Boucadair, M., Levis, P., Bajko, G., Savolainen, T., and T. Tsou, "Huawei Port Range Configuration Options for PPP IP Control Protocol (IPCP)", RFC 6431, November 2011.

13.2. Informative References

[I-D.cui-softwire-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-softwire-b4-translated-ds-lite-10 (work in progress), February 2013.

[I-D.ietf-dhc-dhcpv4-over-ipv6]
Cui, Y., Wu, P., Wu, J., and T. Lemon, "DHCPv4 over IPv6 Transport", draft-ietf-dhc-dhcpv4-over-ipv6-05 (work in progress), September 2012.

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-29 (work in progress), November 2012.

[I-D.ietf-softwire-map]
Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., and T. Murakami, "Mapping of Address and Port with Encapsulation (MAP)", draft-ietf-softwire-map-04 (work in progress), February 2013.

[I-D.ietf-softwire-map-dhcp]
Mrugalski, T., Troan, O., Dec, W., Bao, C., leaf.yeh.sdo@gmail.com, l., and X. Deng, "DHCPv6 Options for Mapping of Address and Port", draft-ietf-softwire-map-dhcp-03 (work in progress), February 2013.

[I-D.ietf-softwire-public-4over6]
Cui, Y., Wu, J., Wu, P., Vautrin, O., and Y. Lee, "Public IPv4 over IPv6 Access Network",

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

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

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

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

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

Authors' Addresses

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

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

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

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

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

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

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

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

Email: yiu_lee@cable.comcast.com

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

Email: ian.farrer@telekom.de

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: November 15, 2013

Y. Fu
S. Jiang
B.Liu
Huawei Technologies Co., Ltd
J.Dong
P. Wu
Tsinghua University
May 14, 2013

Definitions of Managed Objects for MAP-E
draft-fu-softwire-map-mib-05

Status of this Memo

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

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

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

This Internet-Draft will expire on November 15, 2013.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

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

Table of Contents

1. Introduction 3
2. The Internet-Standard Management Framework 3
3. Terminology 3
4. Structure of the MIB Module 3
4.1. The mapMIBObjects 4
4.1.1. The mapRule Subtree 4
4.1.2. The mapSecurityCheck Subtree 4
4.2. The mapMIBConformance Subtree 4
5. Definitions 4
6. IANA Considerations 12
7. Security Considerations 12
8. Acknowledgments 12
9. References 12
9.1. Normative References 12
9.2. Informative References 13
10. Change Log [RFC Editor please remove] 13
Author's Addresses 14

1. Introduction

MAP [I-D. draft-ietf-softwire-map] is a stateless mechanism for running IPv4 over IPv6-only infrastructure. In particular, it includes two mode, translation mode or encapsulation mode. For the encapsulation mode, it provides an automatic tunnelling mechanism for providing IPv4 connectivity service to end users over a service provider's IPv6 network.

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. This MIB module may be used for monitoring the devices in the MAP scenario, especially, for the encapsulation mode.

2. The Internet-Standard Management Framework

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

Managed objects are accessed via a virtual information store, termed the MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP).

Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in [RFC2578], [RFC2579] and [RFC2580].

3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

4. Structure of the MIB Module

The MAP-E MIB provides a way to configure and manage the devices in MAP encapsulation mode through SNMP.

MAP-E MIB is configurable on a per-interface basis. It depends on several parts of the IF-MIB [RFC2863].

4.1. The mapMIBObjects

4.1.1. The mapRule Subtree

The mapRule subtree describes managed objects used for managing the multiple mapping rules in the MAP encapsulation mode.

According to the MAP specification, the mapping rules are divided into two categories, which are BMR (Basic Mapping Rule), and FMR (Forwarding Mapping Rule).

4.1.2. The mapSecurityCheck Subtree

The mapSecurityCheck subtree is to statistic the number of invalid packets that been identified. There are two kind of invalid packets which are defined in the MAP specification as the following.

- The BR MUST perform a validation of the consistency of the source IPv6 address and source port number for the packet using BMR.
- The CE SHOULD check that MAP received packets' transport-layer destination port number is in the range configured by MAP for the CE.

4.2. The mapMIBConformance Subtree

The mapMIBConformance subtree provides conformance information of MIB objects.

5. Definitions

```
MAP-E-MIB DEFINITIONS  ::=  BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, mib-2, transmission,
   Gauge32, Integer32, Counter64
      FROM SNMPv2-SMI  --[RFC2578]

   RowStatus, StorageType, DisplayString
      FROM SNMPv2-TC   --[RFC2579]

   ifIndex, InterfaceIndexOrZero
      FROM IF-MIB      --[RFC2863]

   InetAddressType, InetAddress,
   InetPortNumber, InetAddressPrefixLength
      FROM INET-ADDRESS-MIB  --[RFC4001]
```

```
        OBJECT-GROUP, MODULE-COMPLIANCE,
        NOTIFICATION-GROUP
           FROM SNMPv2-CONF; --[RFC2580]

     mapMIB MODULE-IDENTITY
     LAST-UPDATED "201302070000Z" -- February 6, 2013
     ORGANIZATION "IETF Softwire Working Group"
     CONTACT-INFO
        "Yu Fu
         Huawei Technologies Co., Ltd
         Huawei Building, 156 Beiqing Rd., Hai-Dian District
         Beijing, P.R. China 100095
         EMail: eleven.fuyu@huawei.com

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

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

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

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

     DESCRIPTION
        "The MIB module is defined for management of objects in the
        MAP-E BRs or CEs."

      REVISION    "201305140000Z"
```

```
        ::=  {  transmission xxx  }  --xxx to be replaced with
        IANA-assigned value

       mapMIBObjects OBJECT IDENTIFIER ::= {mapMIB 1}

       mapRule   OBJECT IDENTIFIER
         ::=  { mapMIBObjects 1 }

       mapSecurityCheck   OBJECT IDENTIFIER
         ::=  { mapMIBObjects 2 }

      mapRuleTable OBJECT-TYPE
         SYNTAX     SEQUENCE OF mapRuleEntry
         MAX-ACCESS  read-create
         STATUS     current
         DESCRIPTION
            "The (conceptual) table containing rule Information of
             specific mapping rule. It can also be used for row
             creation."
         ::=  { mapRule 1 }

      mapRuleEntry OBJECT-TYPE
         SYNTAX     MapRuleEntry
         MAX-ACCESS  read-create
         STATUS     current
         DESCRIPTION
            "Each entry in this table contains the information on a
             particular mapping rule."
             INDEX   { mapRuleID }
         ::=  { mapRuleTable 1 }

      mapRuleEntry  ::=
         SEQUENCE {
          mapRuleID                   Integer32,
          mapRuleIPv6PrefixType       InetAddressType,
          mapRuleIPv6Prefix           InetAddress,
          mapRuleIPv6PrefixLen        InetAddressPrefixLength,
          mapRuleIPv4PrefixType       InetAddressType,
          mapRuleIPv4Prefix           InetAddress,
          mapRuleIPv4PrefixLen        InetAddressPrefixLength,
          mapRuleStartPort            InetPortNumber,
          mapRuleEndPort              InetPortNumber,
          mapRuleEALen                Integer32,
          mapRuleStatus               RowStatus,
          mapRuleStorageType          StorageType,
```

```
        mapRuleType                 Integer32
    }

    mapRuleID OBJECT-TYPE
        SYNTAX Integer32 (1..2147483647)
        MAX-ACCESS read-create
        STATUS current
        DESCRIPTION
           "An identifier used to distinguish the multiple mapping
            rule which is unique with each CE in the same BR."
        ::= { mapRuleEntry 1 }

    mapRuleIPv6PrefixType OBJECT-TYPE
        SYNTAX     InetAddressType
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
           "In this object, it MUST be set to the value of 2 to
            present IPv6 type. It complies the textule convention
            of IPv6 address defined in [RFC4001]."
        ::= { mapRuleEntry 2 }

    mapRuleIPv6Prefix OBJECT-TYPE
        SYNTAX     InetAddress
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
           "The IPv6 prefix defined in mapping rule which will be
            assigned to CE ."
        ::= { mapRuleEntry 3 }

    mapRuleIPv6PrefixLen OBJECT-TYPE
        SYNTAX     InetAddressPrefixLength
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
           "The length of the IPv6 prefix defined in the mapping rule.
            As a parameter for mapping rule, it will be also assigned
            to CE."
        ::= { mapRuleEntry 4 }

    mapRuleIPv4PrefixType OBJECT-TYPE
        SYNTAX     InetAddressType
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
           "In this object, it MUST be set to the value of 1 to
```

```
            present IPv4 type. It complies the textual convention
            of IPv6 address defined in [RFC4001]."
        ::= { mapRuleEntry 5 }

    mapRuleIPv4Prefix OBJECT-TYPE
        SYNTAX     InetAddress
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
           " The IPv4 prefix defined in mapping rule which will be
            assigned to CE."
        ::= { mapRuleEntry 6 }

    mapRuleIPv4PrefixLen OBJECT-TYPE
        SYNTAX     InetAddressPrefixLength
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
           "The length of the IPv4 prefix defined in the mapping
            rule. As a parameter for mapping rule, it will be also
            assigned to CE."
        ::= { mapRuleEntry 7 }

    mapRuleStartPort OBJECT-TYPE
        SYNTAX     InetPortNumber
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
           "The start port number of the port range derived
            from the mapping rule which will be assigned to CE."
        ::= { mapRuleEntry 8 }

     mapRuleEndPort OBJECT-TYPE
        SYNTAX     InetPortNumber
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
           " The end port number of the port range derived
            from the mapping rule which will be assigned to CE."
        ::= { mapRuleEntry 9 }

    mapRuleEALen OBJECT-TYPE
        SYNTAX     Integer32
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
           "The length of the Embedded-Address (EA) defined in
```

```
            mapping rule which will be assigned to CE."
        ::= { mapRuleEntry 10 }

    mapRuleStatus OBJECT-TYPE
        SYNTAX     RowStatus
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
           "The status of this row, by which new entries may be
            created, or old entries deleted from this table.
        ::= { mapRuleEntry 11 }

    mapRuleStorageType OBJECT-TYPE
        SYNTAX     StorageType
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
           "The storage type of this row. If the row is
            permanent(4), no objects in the row need be
            writable."
        ::= { mapRuleEntry 12 }

    mapRuleType OBJECT-TYPE
        SYNTAX     Integer32
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
           "The type of the mapping rule. A value of 0 means it
            is a BMR; a non-zero value means it is a FMR."
        ::= { mapRuleEntry 12 }

    mapSecurityCheckTable OBJECT-TYPE
       SYNTAX     SEQUENCE OF MapSecurityCheckEntry
       MAX-ACCESS  read-create
       STATUS     current
       DESCRIPTION
          "The (conceptual) table containing information on
           MAP security checks. This table can be used to statistic
           the number of invalid packets that been identified"
       ::=  { mapSecurityCheck 1 }

    mapSecurityCheckEntry OBJECT-TYPE
       SYNTAX     mapSecurityCheckEntry
       MAX-ACCESS  read-create
       STATUS     current
       DESCRIPTION
          "Each entry in this table contains the information on a
```

```
            particular MAP SecurityCheck."
            INDEX   { mapSecurityCheckInvalidv4,
                      mapSecurityCheckInvalidv6}
         ::=  { mapSecurityCheckTable 1 }

      mapSecurityCheckEntry  ::=
         SEQUENCE {
          mapSecurityCheckInvalidv4       Counter64,
          mapSecurityCheckInvalidv6       Counter64,
          mapSecurityCheckStatus          RowStatus,
          mapSecurityCheckStorageType     StorageType
      }

      mapSecurityCheckInvalidv4 OBJECT-TYPE
          SYNTAX     Counter64
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
             "The CE SHOULD check that MAP received packets'
              transport-layer destination port number is in the range
              configured by MAP for the CE"
          ::= { mapSecurityCheckEntry 1 }

      mapSecurityCheckInvalidv6 OBJECT-TYPE
          SYNTAX     Counter64
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
             "The BR MUST perform a validation of the consistency of
              the source IPv6 address and source port number for the
              packet using BMR."
          ::= { mapSecurityCheckEntry 2 }

      mapSecurityCheckStatus OBJECT-TYPE
          SYNTAX     RowStatus
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
             "The status of this row, by which new entries may be
              created, or old entries deleted from this table.
          ::= { mapSecurityCheckEntry 3 }

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

```
            "The storage type of this row. If the row is
             permanent(4), no objects in the row need be
             writable."
         ::= { mapSecurityCheckEntry 4 }


   -- Conformance Information

      mapMIBConformance OBJECT IDENTIFIER ::= {mapMIB 2}

      mapMIBCompliances OBJECT IDENTIFIER ::= { mapMIBConformance 1 }

      mapMIBGroups OBJECT IDENTIFIER ::= { mapMIBConformance 2 }


   -- compliance statements

      mapMIBCompliance MODULE-COMPLIANCE
         STATUS current
         DESCRIPTION
            " Describes the minimal requirements for conformance
              to the MAP-E MIB."
         MODULE -- this module
             MANDATORY-GROUPS { mapMIBRuleGroup }
        ::= { mapMIBCompliances 1 }

   -- Units of Conformance

      mapMIBRuleGroup OBJECT-GROUP
         OBJECTS { mapRuleBRAddress, mapMapRuleID,
                   mapRuleIPv6Prefix,
                   mapRuleIPv6PrefixLen,
                   mapRuleIPv4Prefix,
                   mapRuleIPv4PrefixLen,
                   mapRuleStartPort,
                   mapRuleEndPort mapRuleEALen,
                   mapRuleStorageType }
         STATUS current
         DESCRIPTION
            " The collection of this objects are used to give the
             information of mapping rules in MAP-E."
         ::= { mapMIBGroups 1 }

             END
```

6. 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
MAP-E-MIB	{ transmission XXX }

7. Security Considerations

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

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 principles (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

8. Acknowledgments

The authors would like to thank for valuable comments from David Harrington, Mark Townsley, and Shishio Tsuchiya.

9. References

9.1. Normative References

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

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

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

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

[RFC2863] McCloghrie, K. and F. Kastenholz. "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", RFC 3411, December 2002.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.

[RFC4087] Thaler, D., "IP Tunnel MIB", RFC 4087, June 2005.

[I-D.ietf-softwire-map]
Troan, O.,etc., "Mapping of Address and Port (MAP)", draft-ietf-softwire-map, working in progress.

[I-D.mdt-softwire-map-dhcp-option]
Mrugalski, T.,etc., "DHCPv6 Options for Mapping of Address and Port", draft-mdt-softwire-map-dhcp-option, working in progress.

9.2. 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.

10. Change Log [RFC Editor please remove]

draft-fu-softwire-map-mib-00, original version, 2012-03-01
draft-fu-softwire-map-mib-01, 01 version, 2012-07-16
draft-fu-softwire-map-mib-03, deleted tunnel object according to the discussion in IETF85, 2013-02-04
draft-fu-softwire-map-mib-04, added security check object according to discussion in IETF86

draft-fu-softwire-map-mib-05, distinguishing FMR and BMR in mapRule object definition; added some description in section 4; modifying a little bit to the mapRuleEntry definiction

Author's Addresses

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

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

Bing Liu
Huawei Technologies Co., Ltd
Huawei Building, 156 Beiqing Rd.,
Hai-Dian District, Beijing 100095
P.R. China
Email: leo.liubing@huawei.com

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

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

Softwire WG
Internet-Draft
Intended status: Standards Track
Expires: December 5, 2015

Q. Wang
China Telecom
W. Meng
C. Wang
ZTE Corporation
M. Boucadair
France Telecom
June 3, 2015

RADIUS Extensions for IPv4-Embedded Multicast and Unicast IPv6 Prefixes
draft-hu-softwire-multicast-radius-ext-08

Abstract

This document specifies a new Remote Authentication Dial-In User Service (RADIUS) attribute to carry the Multicast-Prefixes-64 information, aiming to delivery the Multicast and Unicast IPv6 Prefixes to be used to build multicast and unicast IPv4-Embedded IPv6 addresses. this RADIUS attribute is defined based on the equivalent DHCPv6 OPTION_v6_PREFIX64 option.

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 5, 2015.

Copyright Notice

Copyright (c) 2015 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. Convention and Terminology 4
3. Multicast-Prefixes-64 Configuration with RADIUS and DHCPv6 . . 5
4. RADIUS Attribute 8
4.1. Multicast-Prefixes-64 8
5. Table of Attributes 11
6. Security Considerations 12
7. IANA Considerations 13
8. Acknowledgments 14
9. Normative References 15
Authors' Addresses 16

1. Introduction

The solution specified in [I-D.ietf-softwire-dslite-multicast] relies on stateless functions to graft part of the IPv6 multicast distribution tree and IPv4 multicast distribution tree, also uses IPv4-in-IPv6 encapsulation scheme to deliver IPv4 multicast traffic over an IPv6 multicast-enabled network to IPv4 receivers.

To inform the mB4 element of the PREFIX64,a PREFIX64 option may be used. [I-D.ietf-softwire-multicast-prefix-option] defines a DHCPv6 PREFIX64 option to convey the IPv6 prefixes to be used for constructing IPv4-embedded IPv6 addresses.

In broadband environments, a customer profile may be managed by Authentication, Authorization, and Accounting (AAA) servers, together with AAA for users. The Remote Authentication Dial-In User Service (RADIUS) protocol [RFC2865] is usually used by AAA servers to communicate with network elements. Since the Multicast-Prefixes-64 information can be stored in AAA servers and the client configuration is mainly provided through DHCP running between the NAS and the requesting clients, a new RADIUS attribute is needed to send Multicast-Prefixes-64 information from the AAA server to the NAS.

This document defines a new RADIUS attribute to be used for carrying the Multicast-Prefixes-64, based on the equivalent DHCPv6 option already specified in [I-D.ietf-softwire-multicast-prefix-option].

This document makes use of the same terminology defined in [I-D.ietf-softwire-dslite-multicast].

This attribute can be in particular used in the context of DS-Lite Mulitcast, MAP-E Multicast and other IPv4-IPv6 Multicast techniques. However it is not limited to DS-Lite Multicast.

DS-Lite unicast RADIUS extentions are defined in [RFC6519] .

2. Convention and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The terms DS-Lite multicast Basic Bridging BroadBand element (mB4) and the DS-Lite multicast Address Family Transition Router element (mAFTR) are defined in [I-D.ietf-softwire-dslite-multicast]

3. Multicast-Prefixes-64 Configuration with RADIUS and DHCPv6

Figure 1 illustrates in DS-Lite scenario how the RADIUS protocol and DHCPv6 work together to accomplish Multicast-Prefixes-64 configuration on the mB4 element for multicast service when an IP session is used to provide connectivity to the user.

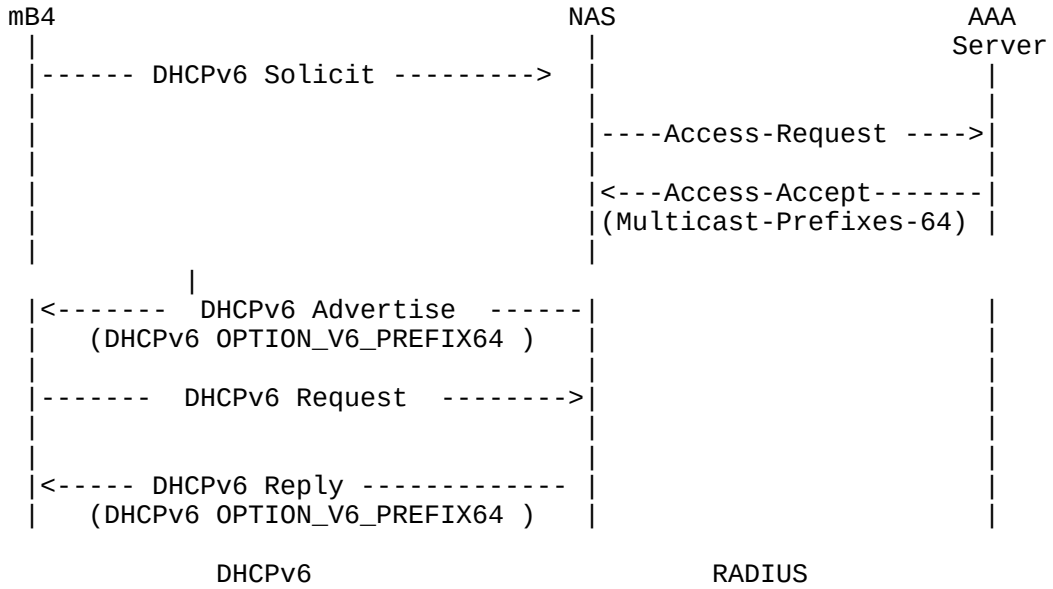


Figure 1: RADIUS and DHCPv6 Message Flow for an IP Session

The NAS operates as a client of RADIUS and as a DHCP Server/Relay for mB4. When the mB4 sends a DHCPv6 Solicit message to NAS(DHCP Server/Relay). The NAS sends a RADIUS Access-Request message to the RADIUS server, requesting authentication. Once the RADIUS server receives the request, it validates the sending client, and if the request is approved, the AAA server replies with an Access-Accept message including a list of attribute-value pairs that describe the parameters to be used for this session. This list MAY contain the Multicast-Prefixes-64 attribute (asm-length,ASM_PREFIX64,ssm-length, SSM_PREFIX64,unicast-length,U_PREFIX64). Then, when the NAS receives the DHCPv6 Request message containing the OPTION_V6_PREFIX64 option in its Option Request option,the NAS SHALL use the prefixes returned in the RADIUS Multicast-Prefixes-64 attribute to populate the DHCPv6 OPTION_V6_PREFIX64 option in the DHCPv6 reply message.

NAS MAY be configured to return the configured Multicast-Prefixes-64 by the AAA Server to any requesting client without relaying each received request to the AAA Server.

Figure 2 describes another scenario, which accomplish DS-Lite Multicast-Prefixes-64 configuration on the mB4 element for multicast service when a PPP session is used to provide connectivity to the user. Once the NAS obtains the Multicast-Prefixes-64 attribute from the AAA server through the RADIUS protocol, the NAS MUST store the received Multicast-Prefixes-64 locally. When a user is online and sends a DHCPv6 Request message containing the OPTION_V6_PREFIX64 option in its Option Request option, the NAS retrieves the previously stored Multicast-Prefixes-64 and uses it as OPTION_V6_PREFIX64 option in DHCPv6 Reply message.

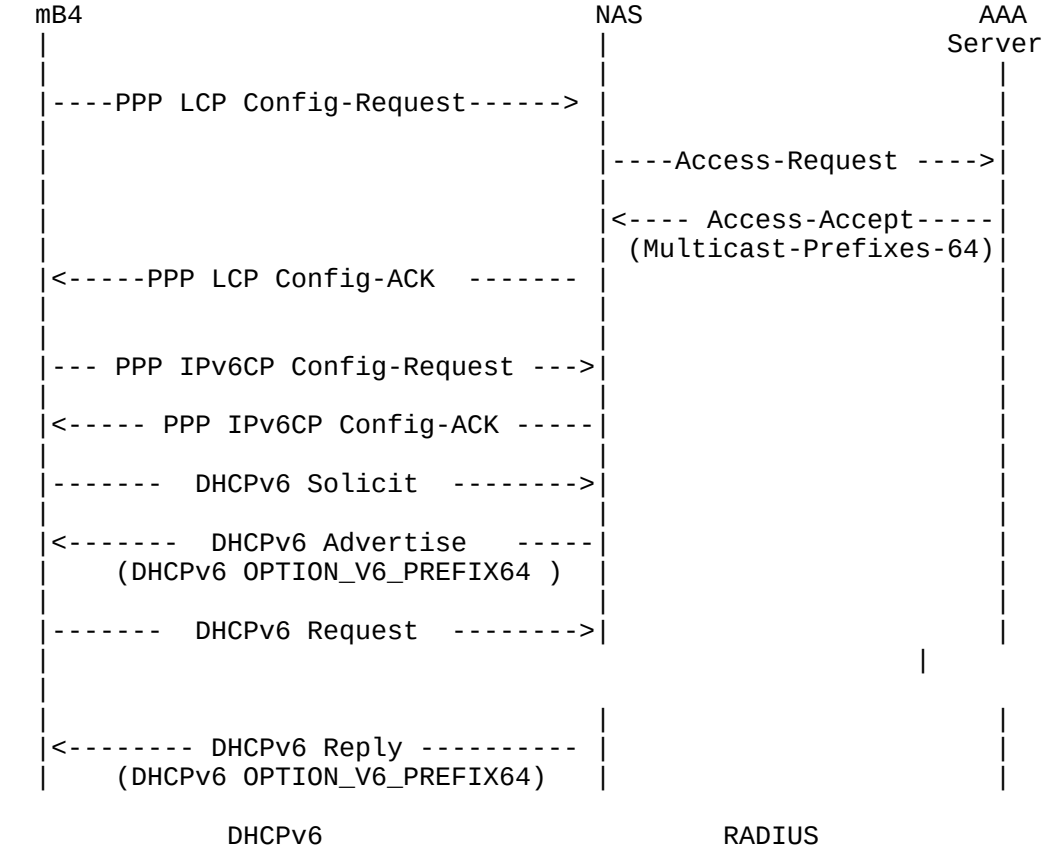


Figure 2: RADIUS and DHCPv6 Message Flow for a PPP Session

According to [RFC3315], after receiving the Multicast-Prefixes-64 attribute in the initial Access-Accept packet, the NAS MUST store the received V6_PREFIX64 locally. When the mB4 sends a DHCPv6 Renew message to request an extension of the lifetimes for the assigned address or prefix, the NAS does not have to initiate a new Access-

Request packet towards the AAA server to request the Multicast-Prefixes-64. The NAS retrieves the previously stored Multicast-Prefixes-64 and uses it in its reply.

Also, if the DHCPv6 server to which the DHCPv6 Renew message was sent at time T1 has not responded, the DHCPv6 client initiates a Rebind/Reply message exchange with any available server. In this scenario, the NAS receiving the DHCPv6 Rebind message MUST initiate a new Access-Request message towards the AAA server. The NAS MAY include the Multicast-Prefixes-64 attribute in its Access-Request message.

4. RADIUS Attribute

This section specifies the format of the new RADIUS attribute.

4.1. Multicast-Prefixes-64

The Multicast-Prefixes-64 attribute conveys the IPv6 prefixes to be used in [I-D.ietf-softwire-dslite-multicast] to synthesize IPv4-embedded IPv6 addresses. The NAS SHALL use the IPv6 prefixes returned in the RADIUS Multicast-Prefixes-64 attribute to populate the DHCPv6 PREFIX64 Option [I-D.ietf-softwire-multicast-prefix-option] .

This attribute MAY be used in Access-Request packets as a hint to the RADIUS server, for example, if the NAS is pre-configured with Multicast-Prefixes-64, these prefixes MAY be inserted in the attribute. The RADIUS server MAY ignore the hint sent by the NAS, and it MAY assign a different Multicast-Prefixes-64 attribute.

If the NAS includes the Multicast-Prefixes-64 attribute, but the AAA server does not recognize this attribute, this attribute MUST be ignored by the AAA server.

NAS MAY be configured with both ASM_PREFIX64 and SSM_PREFIX64 or only one of them. Concretely, AAA server MAY return ASM_PREFIX64 or SSM_PREFIX64 based on the user profile and service policies. AAA MAY return both ASM_PREFIX64 and SSM_PREFIX64. When SSM_PREFIX64 is returned by the AAA server, U_PREFIX64 MUST also be returned by the AAA server.

If the NAS does not receive the Multicast-Prefixes-64 attribute in the Access-Accept message, it MAY fall back to a pre-configured default Multicast-Prefixes-64, if any. If the NAS does not have any pre-configured, the delivery of multicast traffic is not supported.

If the NAS is pre-provisioned with a default Multicast-Prefixes-64 and the Multicast-Prefixes-64 received in the Access-Accept message are different from the configured default, then the Multicast-Prefixes-64 attribute received in the Access-Accept message MUST be used for the session.

A summary of the Multicast-Prefixes-64 RADIUS attribute format is shown Figure 3. The fields are transmitted from left to right.

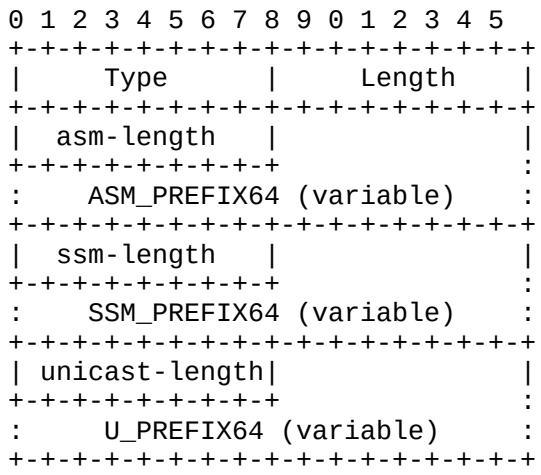


Figure 3: RADIUS attribute format for Multicast-Prefixes-64

Type:

145 for Multicast-Prefixes-64

Length:

This field indicates the total length in octets of this attribute including the Type and Length fields, and the length in octets of all PREFIX fields.

asm-length:

the prefix-length for the ASM IPv4-embedded prefix, as an 8-bit unsigned integer (0 to 128). This field represents the number of valid leading bits in the prefix.

ASM_PREFIX64:

this field identifies the IPv6 multicast prefix to be used to synthesize the IPv4-embedded IPv6 addresses of the multicast groups in the ASM mode. It is a variable size field with the length of the field defined by the asm-length field and is rounded up to the nearest octet boundary. In such case any additional padding bits must be zeroed. The conveyed multicast IPv6 prefix MUST belong to the ASM range. This prefix is likely to be a /96.

ssm-length:

the prefix-length for the SSM IPv4-embedded prefix, as an 8-bit unsigned integer (0 to 128). This field represents the number of valid leading bits in the prefix.

SSM_PREFIX64:

this field identifies the IPv6 multicast prefix to be used to synthesize the IPv4-embedded IPv6 addresses of the multicast groups in the SSM mode. It is a variable size field with the length of the field defined by the ssm-length field and is rounded up to the nearest octet boundary. In such case any additional padding bits must be zeroed. The conveyed multicast IPv6 prefix MUST belong to the SSM range. This prefix is likely to be a /96.

unicast-length:

the prefix-length for the IPv6 unicast prefix to be used to synthesize the IPv4-embedded IPv6 addresses of the multicast sources, as an 8-bit unsigned integer (0 to 128). This field represents the number of valid leading bits in the prefix.

U_PREFIX64:

this field identifies the IPv6 unicast prefix to be used in SSM mode for constructing the IPv4-embedded IPv6 addresses representing the IPv4 multicast sources in the IPv6 domain. U_PREFIX64 may also be used to extract the IPv4 address from the received multicast data flows. It is a variable size field with the length of the field defined by the unicast-length field and is rounded up to the nearest octet boundary. In such case any additional padding bits must be zeroed. The address mapping MUST follow the guidelines documented in [RFC6052].

5. Table of Attributes

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

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

Access-Request	Access-Accept	Access-Reject	Challenge	Accounting-Request	#	Attribute
0-1	0-1	0	0	0-1	145	Multicast-Prefixes-64

CoA-Request	CoA-ACK	CoA-NACK	#	Attribute
0-1	0	0	145	Multicast-Prefixes-64

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

6. Security Considerations

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

The security considerations documented in [RFC3315] and [RFC6052] are to be considered.

7. IANA Considerations

Per this document, IANA has allocated a new RADIUS attribute type from the IANA registry "Radius Attribute Types" located at http://www.iana.org/assignments/radius-types.

Multicast-Prefixes-64 - 145

8. Acknowledgments

The authors would like to thank Ian Farrer, Chongfen Xie, Qi Sun, Linhui Sun and Hao Wang for their contributions to this work.

9. Normative References

[I-D.ietf-softwire-dslite-multicast]
Qin, J., Boucadair, M., Jacquenet, C., Lee, Y., and Q. Wang, "Delivery of IPv4 Multicast Services to IPv4 Clients over an IPv6 Multicast Network", draft-ietf-softwire-dslite-multicast-09 (work in progress), March 2015.

[I-D.ietf-softwire-multicast-prefix-option]
Boucadair, M., Qin, J., Tsou, T., and X. Deng, "DHCPv6 Option for IPv4-Embedded Multicast and Unicast IPv6 Prefixes", draft-ietf-softwire-multicast-prefix-option-08 (work in progress), March 2015.

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

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

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

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

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

[RFC6519] Maglione, R. and A. Durand, "RADIUS Extensions for Dual-Stack Lite", RFC 6519, February 2012.

Authors' Addresses

Qian Wang
China Telecom
No.118, Xizhimennei
Beijing 100035
China

Email: wangqian@ctbri.com.cn

Wei Meng
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing
China

Email: meng.wei2@zte.com.cn,vally.meng@gmail.com

Cui Wang
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing
China

Email: wang.cui1@zte.com.cn

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: July 6, 2016

Y. Fu
CNNIC
S. Jiang
Huawei Technologies Co., Ltd
J. Dong
Y. Chen
Tsinghua University
January 3, 2016

DS-Lite Management Information Base (MIB) for AFTRs
draft-ietf-softwire-dslite-mib-15

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on July 6, 2016.

Copyright Notice

Copyright (c) 2016 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 2
2. Requirements Language 3
3. The Internet-Standard Management Framework 3
4. Relationship to the IF-MIB 3
5. Difference from the IP tunnel MIB and NATV2-MIB 3
6. Structure of the MIB Module 4
6.1. The Object Group 5
6.1.1. The dsliteTunnel Subtree 5
6.1.2. The dsliteNAT Subtree 5
6.1.3. The dsliteInfo Subtree 5
6.2. The Notification Group 5
6.3. The Conformance Group 5
7. MIB modules required for IMPORTS 5
8. Definitions 6
9. Security Considerations 22
10. IANA Considerations 23
11. Acknowledgements 24
12. References 24
12.1. Normative References 24
12.2. Informative References 25
Authors' Addresses 26

1. Introduction

Dual-Stack Lite [RFC6333] is a solution to offer both IPv4 and IPv6 connectivity to customers crossing an IPv6 only infrastructure. One of its key components is an IPv4-over-IPv6 tunnel, which is used to provide IPv4 connectivity across a service provider's IPv6 network. Another key component is a carrier-grade IPv4-IPv4 Network Address Translation (NAT) to share service provider IPv4 addresses among customers.

This document defines a portion of the Management Information Base (MIB) for using with network management protocols in the Internet community. This MIB module may be used for configuration and monitoring Address Family Transition Routers (AFTRs) in a Dual-Stack Lite scenario.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings, and are not to be interpreted as [RFC2119] key words.

3. The Internet-Standard Management Framework

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

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

4. Relationship to the IF-MIB

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

The ifOperStatus in ifTable is used to represent whether the DS-Lite tunnel function has been triggered. The ifInUcastPkts defined in ifTable will represent the number of IPv4 packets that have been encapsulated into IPv6 packets sent to a B4. The ifOutUcastPkts defined in ifTable contains the number of IPv6 packets that can be decapsulated to IPv4 in the virtual interface. Also, the IF-MIB defines ifMtu for the MTU of this tunnel interface, so DS-Lite MIB does not need to define the MTU for the tunnel.

5. Difference from the IP tunnel MIB and NATV2-MIB

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

Notes: According to section 5.2 of [RFC6333], DS-Lite only defines IPv4 in IPv6 tunnels at this moment, but other types of encapsulation could be defined in the future. So this DS-Lite MIB only supports IP in IP encapsulation. If another RFC defines other tunnel types in the future, this DS-Lite MIB will be updated then.

The NATV2-MIB [RFC7659] is designed to carry translation from any address family to any address family, therefore it supports IPv4 to IPv4 translation.

The IP Tunnel MIB [RFC4087] is designed for managing tunnels of any type over IPv4 and IPv6 networks, therefore it has already supports IP in IP tunnels. But in a DS-Lite scenario, the tunnel type is point-to-multipoint IP in IP tunnels. The direct(2) defined in IP Tunnel MIB only supports point-to-point tunnel. So it needs to define a new tunnel type for DS-Lite.

However, the NATV2-MIB and IP Tunnel MIB together are not sufficient to support DS-Lite. This document describes the specific features for DS-Lite MIB, as below.

In the DS-Lite scenario, the Address Family Transition Router (AFTR) is not only the tunnel end concentrator, but also an IPv4-to-IPv4 NAT. So as defined in [RFC6333], when the IPv4 packets come back from the Internet to the AFTR, it knows how to reconstruct the IPv6 encapsulation by doing a reverse lookup in the extended IPv4 NAT binding table (section 6.6 of [RFC6333]). The NAT binding table in the AFTR is extended to include the IPv6 address of the tunnel initiator. However, the NAT binding information defined in NATV2-MIB as natv2PortMapTable is indexed by the NAT instance, protocol, and external realm and address. Because the tunnelIfTable defined in the TUNNEL-MIB [RFC4087] is indexed by the ifIndex, the DS-Lite-MIB needs to define the tunnel objects to extend the NAT binding entry by interface. Therefore, a combined MIB is necessary.

An implementation of the IP Tunnel MIB is required for DS-Lite. As the tunnel is not point-to-point in DS-Lite, it needs to define a new tunnel type for DS-Lite. And the tunnelIfEncapsMethod in the tunnelIfEntry should be set to dsLite ("xx"), and a corresponding entry in the DS-Lite module will exist for every tunnelIfEntry with this tunnelIfEncapsMethod. The tunnelIfRemoteInetAddress must be set to "::".

6. Structure of the MIB Module

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

The DS-Lite MIB is configurable on a per-interface basis. It depends on several parts of the IF-MIB [RFC2863], IP Tunnel MIB [RFC4087], and NATV2-MIB [RFC7659].

6.1. The Object Group

This group defines objects that are needed for DS-Lite MIB.

6.1.1. The dsliteTunnel Subtree

The dsliteTunnel subtree describes managed objects used for managing tunnels in the DS-Lite scenario. Because the tunnelInetConfigLocalAddress and tunnelInetConfigRemoteAddress defined in the IP Tunnel MIB are not readable, a few new objects are defined in DS-Lite MIB.

6.1.2. The dsliteNAT Subtree

The dsliteNAT subtree describes managed objects used for configuration as well as monitoring of an AFTR which is capable of a NAT function. Because the NATV2-MIB supports the NAT management function in DS-Lite, we may reuse it in DS-Lite MIB. The dsliteNAT subtree also provides the mapping information between the tunnel entry (dsliteTunnelEntry) and the NAT entry (dsliteNATBindEntry) by adding the IPv6 address of the B4 to the natv2PortMapEntry in the NATV2-MIB.

6.1.3. The dsliteInfo Subtree

The dsliteInfo subtree provides statistical information for DS-Lite.

6.2. The Notification Group

This group defines some notification objects for a DS-Lite scenario.

6.3. The Conformance Group

The dsliteConformance subtree provides conformance information of MIB objects.

7. MIB modules required for IMPORTS

This MIB module IMPORTs objects from [RFC2578], [RFC2580], [RFC2863], [RFC3411], [RFC4001] and [RFC7659].

8. Definitions

```
 DSLite-MIB DEFINITIONS  ::=  BEGIN

      IMPORTS
         MODULE-IDENTITY, OBJECT-TYPE, mib-2,
         NOTIFICATION-TYPE,  Integer32,
         Counter64, Unsigned32
            FROM SNMPv2-SMI

         OBJECT-GROUP, MODULE-COMPLIANCE,
         NOTIFICATION-GROUP
            FROM SNMPv2-CONF

         SnmpAdminString
            FROM SNMP-FRAMEWORK-MIB

         ifIndex
            FROM IF-MIB

        InetAddress, InetAddressType, InetAddressPrefixLength,
        InetPortNumber
              FROM INET-ADDRESS-MIB

        ProtocolNumber, Natv2InstanceIndex, Natv2SubscriberIndex
             FROM NATV2-MIB;

      dsliteMIB MODULE-IDENTITY
      LAST-UPDATED "201601030000Z"          -- January 03, 2016
      ORGANIZATION "IETF Softwire Working Group"
      CONTACT-INFO
         "Yu Fu
          CNNIC
          No.4 South 4th Street, Zhongguancun, Hai-Dian District
          Beijing, P.R. China 100090
          EMail: fuyu@cnnic.cn

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

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

Email: knight.dongjiang@gmail.com

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

```
      DESCRIPTION
         "The MIB module is defined for management of objects in the
         DS-Lite scenario.
         Copyright (C) The Internet Society (2016).  This version
         of this MIB module is part of RFC yyyy; see the RFC itself
         for full legal notices. "
      REVISION    "201601030000Z"
      DESCRIPTION
         "Initial version. Published as RFC xxxx."
   --RFC Ed.: RFC-edtitor pls fill in xxxx
         ::=  {  mib-2 xxx  }
   --RFC Ed.: assigned by IANA, see section 10 for details



   --Top level components of this MIB module

      dsliteMIBObjects OBJECT IDENTIFIER
         ::=  { dsliteMIB 1 }
      dsliteTunnel   OBJECT IDENTIFIER
         ::=  { dsliteMIBObjects 1 }

      dsliteNAT   OBJECT IDENTIFIER
         ::=  { dsliteMIBObjects 2 }

      dsliteInfo   OBJECT IDENTIFIER
         ::=  { dsliteMIBObjects 3 }

   --Notifications section

      dsliteNotifications  OBJECT IDENTIFIER
         ::=  { dsliteMIB 0 }


   --dsliteTunnel

   --dsliteTunnelTable

      dsliteTunnelTable OBJECT-TYPE
```

```
      SYNTAX     SEQUENCE OF DsliteTunnelEntry
      MAX-ACCESS  not-accessible
      STATUS     current
      DESCRIPTION
         "The (conceptual) table containing information on
          configured tunnels. This table can be used to map
          a B4 address to the associated AFTR address. It can
          also be used for row creation."
      REFERENCE
         "B4, AFTR: RFC6333."
      ::=  { dsliteTunnel 1 }

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

   DsliteTunnelEntry  ::=
      SEQUENCE {
       dsliteTunnelAddressType         InetAddressType,
       dsliteTunnelStartAddress        InetAddress,
       dsliteTunnelEndAddress          InetAddress,
       dsliteTunnelStartAddPreLen      InetAddressPrefixLength
   }

    dsliteTunnelAddressType OBJECT-TYPE
       SYNTAX     InetAddressType
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
          "This object MUST be set to the value of ipv6(2).
           It describes the address type of the IPv4-in-IPv6
           tunnel initiator and endpoint."
       REFERENCE
         "ipv6(2): RFC4001."
       ::= { dsliteTunnelEntry 1 }

   dsliteTunnelStartAddress OBJECT-TYPE
       SYNTAX     InetAddress (SIZE (0..16))
       MAX-ACCESS not-accessible
```

```
          STATUS     current
          DESCRIPTION
             "The IPv6 address of the initiator of the tunnel
              The address type is given by dsliteTunnelAddressType."
          ::= { dsliteTunnelEntry 2 }

      dsliteTunnelEndAddress OBJECT-TYPE
          SYNTAX     InetAddress (SIZE (0..16))
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
             "The IPv6 address of the endpoint of the tunnel
              The address type is given by dsliteTunnelAddressType."
          ::= { dsliteTunnelEntry 3 }

      dsliteTunnelStartAddPreLen OBJECT-TYPE
          SYNTAX InetAddressPrefixLength
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
             "The IPv6 prefix length of the IP address for the
              initiator of the tunnel(dsliteTunnelStartAddress)."
          ::= { dsliteTunnelEntry 4 }


    --dsliteNATBindTable(according to the NAPT scheme)

       dsliteNATBindTable OBJECT-TYPE
         SYNTAX     SEQUENCE OF DsliteNATBindEntry
         MAX-ACCESS not-accessible
         STATUS     current
         DESCRIPTION
            "This table contains information about currently
             active NAT binds in the NAT of the AFTR. This table
             adds the IPv6 address of a B4 to the natv2PortMapTable
             defined in NATV2-MIB (RFC7659)."
         REFERENCE
              "NATV2-MIB: section 4 of RFC7659."
         ::=  { dsliteNAT 1 }

      dsliteNATBindEntry OBJECT-TYPE
         SYNTAX     DsliteNATBindEntry
         MAX-ACCESS not-accessible
         STATUS     current
         DESCRIPTION
            "The entry in this table holds the mapping relationship
             between tunnel information and NAT bind information.
             Each entry in this table not only need to match a
```

```
            corresponding entry in the natv2PortMapTable but
            also a corresponding entry in the dsliteTunnelTable.
            So the INDEX of the entry needs to match a corresponding
            value in the natv2PortMapTable INDEX and a correspongding
            value in the dsliteTunnelTable INDEX. These entries are
            lost upon agent restart."
        REFERENCE
             "natv2PortMapTable: section 4 of RFC7659."
        INDEX   { dsliteNATBindMappingInstanceIndex,
                  dsliteNATBindMappingProto,
                  dsliteNATBindMappingExtRealm,
                  dsliteNATBindMappingExtAddressType,
                  dsliteNATBindMappingExtAddress,
                  dsliteNATBindMappingExtPort,
                  ifIndex,
                  dsliteTunnelStartAddress }
        ::=  {  dsliteNATBindTable 1   }

    DsliteNATBindEntry  ::=
        SEQUENCE {
        dsliteNATBindMappingInstanceIndex  Natv2InstanceIndex,
        dsliteNATBindMappingProto          ProtocolNumber,
        dsliteNATBindMappingExtRealm       SnmpAdminString,
        dsliteNATBindMappingExtAddressType InetAddressType,
        dsliteNATBindMappingExtAddress     InetAddress,
        dsliteNATBindMappingExtPort        InetPortNumber,
        dsliteNATBindMappingIntRealm       SnmpAdminString,
        dsliteNATBindMappingIntAddressType InetAddressType,
        dsliteNATBindMappingIntAddress     InetAddress,
        dsliteNATBindMappingIntPort        InetPortNumber,
        dsliteNATBindMappingPool           Unsigned32,
        dsliteNATBindMappingMapBehavior    INTEGER,
        dsliteNATBindMappingFilterBehavior INTEGER,
        dsliteNATBindMappingAddressPooling INTEGER
        }

     dsliteNATBindMappingInstanceIndex OBJECT-TYPE
         SYNTAX Natv2InstanceIndex
         MAX-ACCESS not-accessible
         STATUS current
         DESCRIPTION
            "Index of the NAT instance that created this port
             map entry."
      ::= { dsliteNATBindEntry 1 }

     dsliteNATBindMappingProto OBJECT-TYPE
         SYNTAX      ProtocolNumber
         MAX-ACCESS  not-accessible
```

```
          STATUS      current
          DESCRIPTION
             "This object specifies the mapping's transport protocol
              number."
          ::= { dsliteNATBindEntry 2 }

      dsliteNATBindMappingExtRealm OBJECT-TYPE
          SYNTAX     SnmpAdminString (SIZE(0..32))
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
             "The realm to which dsliteNATBindMappingExtAddress
             belongs."
          ::= { dsliteNATBindEntry 3 }

       dsliteNATBindMappingExtAddressType OBJECT-TYPE
          SYNTAX InetAddressType
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
            "Address type for the mapping's external address.
             This object MUST be set to the value of iPv4(1).
             The values of ipv6(2), ipv4z(3) and ipv6z(4) are
             not allowed."
          REFERENCE
            "ipv4(1), ipv6(2), iPv4z(3) and ipv6z(4): RFC4001."
          ::= { dsliteNATBindEntry 4 }

      dsliteNATBindMappingExtAddress OBJECT-TYPE
          SYNTAX InetAddress (SIZE (0..4))
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
            "The mapping's external address. This is the source
             address for translated outgoing packets. The address
             type is given by dsliteNATBindMappingExtAddressType."
          ::= { dsliteNATBindEntry 5 }

       dsliteNATBindMappingExtPort OBJECT-TYPE
          SYNTAX InetPortNumber
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
            "The mapping's assigned external port number.
             This is the source port for translated outgoing
             packets. This MUST be a non-zero value."
          ::= { dsliteNATBindEntry 6 }
```

```
       dsliteNATBindMappingIntRealm OBJECT-TYPE
          SYNTAX SnmpAdminString (SIZE(0..32))
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
            "The realm to which natMappingIntAddress belongs. This
             realm defines the IPv6 address space from which the
             tunnel source address is taken. The realm of the
             encapsulated IPv4 address is restricted in scope to
             the tunnel, so there is no point in identifying it
             separately."
          ::= { dsliteNATBindEntry 7 }

    dsliteNATBindMappingIntAddressType OBJECT-TYPE
          SYNTAX InetAddressType
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
             "Address type of the mapping's internal address.
             This object MUST be set to the value of iPv4z(3).
             The values of ipv4(1), ipv6(2) and ipv6z(4) are
             not allowed."
          REFERENCE
            "ipv4(1), ipv6(2), iPv4z(3) and ipv6z(4): RFC4001."
          ::= { dsliteNATBindEntry 8 }

      dsliteNATBindMappingIntAddress OBJECT-TYPE
          SYNTAX InetAddress
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
            "The mapping's internal address. It is the IPv6 tunnel
             source address. The address type is given by
             dsliteNATBindMappingIntAddressType."
          ::= { dsliteNATBindEntry 9 }

      dsliteNATBindMappingIntPort OBJECT-TYPE
          SYNTAX InetPortNumber
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
           "The mapping's internal port number. This MUST be a non-zero
            value."
           ::= { dsliteNATBindEntry 10 }

      dsliteNATBindMappingPool OBJECT-TYPE
          SYNTAX Unsigned32 (0|1..4294967295)
          MAX-ACCESS read-only
```

```
          STATUS current
          DESCRIPTION
            "Index of the pool that contains this mapping's external
             address and port. If zero, no pool is associated with this
             mapping."
          ::= { dsliteNATBindEntry 11 }

      dsliteNATBindMappingMapBehavior OBJECT-TYPE
          SYNTAX INTEGER{
          endpointIndependent (0),
          addressDependent(1),
          addressAndPortDependent (2)
          }
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
            "Mapping behavior as described in [RFC4787] section 4.1.

             endpointIndependent(0), the behavior REQUIRED by
             RFC4787, REQ-1, maps the source address and port to
             the same external address and port for all destination
             address and port combinations reached through the same
             external realm and using the given protocol.

             addressDependent(1) maps to the same external address
             and port for all destination ports at the same
             destination address reached through the same external
             realm and using the given protocol.

             addressAndPortDependent(2) maps to a separate external
             address and port combination for each different
             destination address and port combination reached
             through the same external realm.

             For the DS-Lite scenario, it must be
             addressAndPortDependent(2)."
          REFERENCE
            "Mapping behavior: section 4.1 of RFC4787.
             DS-Lite: RFC 6333."
          ::= { dsliteNATBindEntry 12 }

      dsliteNATBindMappingFilterBehavior OBJECT-TYPE
          SYNTAX INTEGER{
          endpointIndependent (0),
          addressDependent(1),
          addressAndPortDependent (2)
          }
          MAX-ACCESS read-only
```

```
          STATUS current
          DESCRIPTION
            "Filtering behavior as described in [RFC4787] section 5.

             endpointIndependent(0) accepts for translation packets
             from all combinations of remote address and port
             destined to the mapped external address and port via
             the given external realm and using the given protocol.

             addressDependent(1) accepts for translation packets from
             all remote ports from the same remote source address
             destined to the mapped external address and port via the
             given external realm and using the given protocol.

             addressAndPortDependent(2) accepts for translation only
             those packets with the same remote source address, port,
             and protocol incoming from the same external realm as
             identified when the applicable port map entry was
             created.

             RFC 4787, REQ-8 recommends either endpointIndependent(0)
             or addressDependent(1) filtering behavior depending on
             whether application friendliness or security takes
             priority.

             For the DS-Lite scenario, it must be
             addressAndPortDependent(2)."
          REFERENCE
            "Filtering behavior: section 5 of RFC4787.
             DS-Lite: RFC6333."
          ::= { dsliteNATBindEntry 13 }

      dsliteNATBindMappingAddressPooling OBJECT-TYPE
          SYNTAX INTEGER{
          arbitrary (0),
          paired (1)
          }
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
            "Type of address pooling behavior that was used to create
             this mapping.

             arbitrary(0) pooling behavior means that the NAT instance
             may create the new port mapping using any address in the
             pool that has a free port for the protocol concerned.

             paired(1) pooling behavior, the behavior RECOMMENDED by RFC
```

```
            4787, REQ-2, means that once a given internal address has
            been mapped to a particular address in a particular pool,
            further mappings of the same internal address to that pool
            will reuse the previously assigned pool member address."
         REFERENCE
           "Pooling behavior: section 4.1 of RFC4787."
         ::= { dsliteNATBindEntry 14 }


      --dsliteInfo

      dsliteAFTRAlarmScalar OBJECT IDENTIFIER ::= { dsliteInfo 1 }

      dsliteAFTRAlarmB4AddrType OBJECT-TYPE
         SYNTAX  InetAddressType
         MAX-ACCESS accessible-for-notify
         STATUS current
         DESCRIPTION
            "This object indicates the address type of
             the B4 which will send an alarm."
         ::= { dsliteAFTRAlarmScalar 1 }

      dsliteAFTRAlarmB4Addr OBJECT-TYPE
         SYNTAX  InetAddress
         MAX-ACCESS accessible-for-notify
         STATUS current
         DESCRIPTION
            "This object indicates the IP address of
             B4 which will send an alarm. The address type is
             given by dsliteAFTRAlarmB4AddrType."
         ::= { dsliteAFTRAlarmScalar 2 }

      dsliteAFTRAlarmProtocolType OBJECT-TYPE
         SYNTAX INTEGER{
         tcp (0),
         udp (1),
         icmp (2),
         total (3)
         }
         MAX-ACCESS accessible-for-notify
         STATUS current
         DESCRIPTION
            "This object indicates the transport protocol type
             of alarm.

             tcp (0) means that the transport protocoal type of
             alarm is tcp.
```

```
            udp (1) means that the transport protocoal type of
            alarm is udp.

            icmp (2) means that the transport protocoal type of
            alarm is icmp.

            total (3) means that the transport protocoal type of
            alarm is total."
        ::= { dsliteAFTRAlarmScalar 3 }

    dsliteAFTRAlarmSpecificIPAddrType OBJECT-TYPE
        SYNTAX InetAddressType
        MAX-ACCESS accessible-for-notify
        STATUS current
        DESCRIPTION
           "This object indicates the address type of the IP address
            whose port usage has reached the threshold."
        ::= { dsliteAFTRAlarmScalar 4 }

     dsliteAFTRAlarmSpecificIP OBJECT-TYPE
        SYNTAX InetAddress
        MAX-ACCESS accessible-for-notify
        STATUS current
        DESCRIPTION
           "This object indicates the IP address whose port usage
            has reached the threshold. The address type is given by
            dsliteAFTRAlarmSpecificIPAddrType."
        ::= { dsliteAFTRAlarmScalar 5 }

      dsliteAFTRAlarmConnectNumber OBJECT-TYPE
        SYNTAX Integer32 (60..90)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
           "This object indicates the notification threshold
            of the DS-Lite tunnels which is active in
            the AFTR device."
         REFERENCE
           "AFTR: section 6 of RFC6333."
        DEFVAL
            { 60 }
        ::= { dsliteAFTRAlarmScalar 6 }

      dsliteAFTRAlarmSessionNumber OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
```

```
            "This object indicates the notification threshold of
             the IPv4 session for the user."
          REFERENCE
            "AFTR: section 6 of RFC6333
             B4: section 5 of RFC6333."
         DEFVAL
             { -1 }
         ::= { dsliteAFTRAlarmScalar 7 }

      dsliteAFTRAlarmPortNumber OBJECT-TYPE
         SYNTAX Integer32
         MAX-ACCESS read-write
         STATUS current
         DESCRIPTION
            "This object indicates the notification threshold of the NAT
             ports which have been used by user."
         DEFVAL
             { -1 }
         ::= { dsliteAFTRAlarmScalar 8 }

       dsliteStatisticsTable OBJECT-TYPE
         SYNTAX SEQUENCE OF DsliteStatisticsEntry
         MAX-ACCESS not-accessible
         STATUS current
         DESCRIPTION
            "This table provides statistical information
             about DS-Lite."
         ::= { dsliteInfo 2 }

       dsliteStatisticsEntry OBJECT-TYPE
         SYNTAX DsliteStatisticsEntry
         MAX-ACCESS not-accessible
         STATUS current
         DESCRIPTION
            "Ench entry in this table provides statistical information
             about DS-Lite."
         INDEX { dsliteStatisticsSubscriberIndex }
         ::= { dsliteStatisticsTable 1 }

       DsliteStatisticsEntry ::=
         SEQUENCE {
        dsliteStatisticsSubscriberIndex        Natv2SubscriberIndex,
        dsliteStatisticsDiscards               Counter64,
        dsliteStatisticsSends                  Counter64,
        dsliteStatisticsReceives               Counter64,
        dsliteStatisticsIpv4Session            Counter64,
        dsliteStatisticsIpv6Session            Counter64
       }
```

```
     dsliteStatisticsSubscriberIndex OBJECT-TYPE
        SYNTAX Natv2SubscriberIndex
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
           "Index of the subscriber or host. A unique value,
            greater than zero, for each subscriber in the
            managed system."
         ::= { dsliteStatisticsEntry 1 }

      dsliteStatisticsDiscards OBJECT-TYPE
         SYNTAX Counter64
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
            "This object indicates the number of packets
             discarded from this subscriber."
        ::= { dsliteStatisticsEntry 2 }

       dsliteStatisticsSends  OBJECT-TYPE
         SYNTAX Counter64
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
            "This object indicates the number of packets which is
              sent to this subscriber."
         ::= { dsliteStatisticsEntry 3 }

       dsliteStatisticsReceives  OBJECT-TYPE
         SYNTAX Counter64
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
            "This object indicates the number of packets which is
              received from this subscriber."
         ::= { dsliteStatisticsEntry 4 }

      dsliteStatisticsIpv4Session OBJECT-TYPE
         SYNTAX Counter64
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
            "This object indicates the number of the
             current IPv4 Sessions."
         REFERENCE
             "Session: the paragraph 2 of RFC6333 section 11.
             (The AFTR should have the capability to log the
              tunnel-id, protocol, ports/IP addresses, and
```

```
                the creation time of the NAT binding to uniquely
                identify the user sessions)."
           ::= { dsliteStatisticsEntry 5 }

      dsliteStatisticsIpv6Session OBJECT-TYPE
         SYNTAX Counter64
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
            "This object indicates the number of the
             current IPv6 Session. Because the AFTR is
             also a dual-stack device, it will also
             forward normal IPv6 packets for the
             inbound and outbound direction."
         REFERENCE
             "Session: the paragraph 2 of RFC6333 section 11.
             (The AFTR should have the capability to log the
              tunnel-id, protocol, ports/IP addresses, and
              the creation time of the NAT binding to uniquely
              identify the user sessions)."
        ::= { dsliteStatisticsEntry 6 }

   ---dslite Notifications

      dsliteTunnelNumAlarm NOTIFICATION-TYPE
         OBJECTS { dsliteAFTRAlarmProtocolType,
                   dsliteAFTRAlarmB4AddrType,
                   dsliteAFTRAlarmB4Addr }
         STATUS current
         DESCRIPTION
            "This trap is triggered when the number of
             current dslite tunnels exceeds the value of
             dsliteAFTRAlarmConnectNumber."
         ::= { dsliteNotifications 1 }

      dsliteAFTRUserSessionNumAlarm NOTIFICATION-TYPE
         OBJECTS { dsliteAFTRAlarmProtocolType,
                   dsliteAFTRAlarmB4AddrType,
                   dsliteAFTRAlarmB4Addr }
         STATUS current
         DESCRIPTION
            "This trap is triggered when user sessions
             reach the threshold. The threshold
             is specified by the dsliteAFTRAlarmSessionNumber."
         REFERENCE
             "Session: the paragraph 2 of RFC6333 section 11.
             (The AFTR should have the capability to log the
              tunnel-id, protocol, ports/IP addresses, and
```

```
              the creation time of the NAT binding to uniquely
              identify the user sessions)."
          ::= { dsliteNotifications 2 }

        dsliteAFTRPortUsageOfSpecificIpAlarm NOTIFICATION-TYPE
          OBJECTS { dsliteAFTRAlarmSpecificIPAddrType,
                    dsliteAFTRAlarmSpecificIP }
          STATUS current
          DESCRIPTION
             "This trap is triggered when the used NAT
              ports of map address reach the threshold.
              The threshold is specified by the
              dsliteAFTRAlarmPortNumber."
          ::= { dsliteNotifications 3 }

   --Module Conformance statement

      dsliteConformance   OBJECT IDENTIFIER
         ::=  { dsliteMIB 2 }

      dsliteCompliances OBJECT IDENTIFIER ::= { dsliteConformance 1 }

      dsliteGroups OBJECT IDENTIFIER ::= { dsliteConformance 2 }

   -- compliance statements

      dsliteCompliance MODULE-COMPLIANCE
          STATUS current
          DESCRIPTION
              "Describes the minimal requirements for conformance
               to the DSLite-MIB."
          MODULE -- this module
              MANDATORY-GROUPS { dsliteNATBindGroup,
                  dsliteTunnelGroup,
                  dsliteStatisticsGroup,
                  dsliteNotificationsGroup,
                  dsliteAFTRAlarmScalarGroup }
          ::= { dsliteCompliances 1 }

      dsliteNATBindGroup OBJECT-GROUP
         OBJECTS {
                  dsliteNATBindMappingIntRealm,
                  dsliteNATBindMappingIntAddressType,
                  dsliteNATBindMappingIntAddress,
                  dsliteNATBindMappingIntPort,
                  dsliteNATBindMappingPool,
                  dsliteNATBindMappingMapBehavior,
                  dsliteNATBindMappingFilterBehavior,
```

```
                dsliteNATBindMappingAddressPooling }
        STATUS current
        DESCRIPTION
            "A collection of objects to support basic
             management of NAT binds in the NAT of the AFTR."
         ::= { dsliteGroups 1 }

    dsliteTunnelGroup OBJECT-GROUP
       OBJECTS { dsliteTunnelStartAddPreLen }
       STATUS current
       DESCRIPTION
           "A collection of objects to support management
            of ds-lite tunnels."
       ::= { dsliteGroups 2 }

    dsliteStatisticsGroup OBJECT-GROUP
       OBJECTS { dsliteStatisticsDiscards,
                 dsliteStatisticsSends,
                 dsliteStatisticsReceives,
                 dsliteStatisticsIpv4Session,
                 dsliteStatisticsIpv6Session }
      STATUS current
      DESCRIPTION
         " A collection of objects to support management
           of statistical information for AFTR devices."
        ::= { dsliteGroups 3 }

    dsliteNotificationsGroup NOTIFICATION-GROUP
       NOTIFICATIONS { dsliteTunnelNumAlarm,
                       dsliteAFTRUserSessionNumAlarm,
                       dsliteAFTRPortUsageOfSpecificIpAlarm }
       STATUS current
       DESCRIPTION
          "A collection of objects to support management
           of trap information for AFTR devices."
       ::= { dsliteGroups 4 }

     dsliteAFTRAlarmScalarGroup OBJECT-GROUP
       OBJECTS { dsliteAFTRAlarmB4AddrType,
                 dsliteAFTRAlarmB4Addr,
                 dsliteAFTRAlarmProtocolType,
                 dsliteAFTRAlarmSpecificIPAddrType,
                 dsliteAFTRAlarmSpecificIP,
                 dsliteAFTRAlarmConnectNumber,
                 dsliteAFTRAlarmSessionNumber,
                 dsliteAFTRAlarmPortNumber}
       STATUS current
       DESCRIPTION
```

```
            "A collection of objects to surpport management of
            the information about AFTR alarming Scalar."
        ::= { dsliteGroups 5 }

            END
```

9. Security Considerations

There are three objects defined in this MIB module with a MAX-ACCESS clause of read-write. 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 opens devices to attack. These are the tables and objects and their sensitivity/vulnerability:

Notification thresholds: An attacker setting an arbitrarily low threshold can cause many useless notifications to be generated. Setting an arbitrarily high threshold can effectively disable notifications, which could be used to hide another attack.

dsliteAFTRAlarmConnectNumber

dsliteAFTRAlarmSessionNumber

dsliteAFTRAlarmPortNumber

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:

Objects that reveal host identities: Various objects can reveal the identity of private hosts that are engaged in a session with external end nodes. A curious outsider could monitor these to assess the number of private hosts being supported by the AFTR device. Further, a disgruntled former employee of an enterprise could use the information to break into specific private hosts by intercepting the existing sessions or originating new sessions into the host. If nothing else, unauthorized monitoring of these objects will violate individual subscribers' privacy.

entries in dsliteTunnelTable

entries in dsliteNATBindTable

Unauthorized read access to the dsliteTunnelTable would reveal information about the tunnel topology.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), 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.

Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

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.

10. IANA Considerations

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

Descriptor	OBJECT IDENTIFIER value
DSLite-MIB	{ mib-2 XXX }

```
IANAtunnelType ::= TEXTUAL-CONVENTION

            SYNTAX     INTEGER {

                       dsLite ("XX")        -- dslite tunnel

                       }
```

11. Acknowledgements

The authors would like to thanks the valuable comments made by Suresh Krishnan, Ian Farrer, Yiu Lee, Qi Sun, Yong Cui, David Harrington, Dave Thaler, Tassos Chatzithomaoglou, Tom Taylor, Hui Deng, Carlos Pignataro, Matt Miller, Terry Manderson and other members of The SOFTWIRE WG.

This document was produced using the xml2rfc tool [RFC2629].

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <http://www.rfc-editor.org/info/rfc2578>.

[RFC2580] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance Statements for SMIv2", STD 58, RFC 2580, DOI 10.17487/RFC2580, April 1999, <http://www.rfc-editor.org/info/rfc2580>.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <http://www.rfc-editor.org/info/rfc2863>.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, DOI 10.17487/RFC3411, December 2002, <http://www.rfc-editor.org/info/rfc3411>.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, DOI 10.17487/RFC4001, February 2005, <http://www.rfc-editor.org/info/rfc4001>.

[RFC4087] Thaler, D., "IP Tunnel MIB", RFC 4087, DOI 10.17487/RFC4087, June 2005, <http://www.rfc-editor.org/info/rfc4087>.

[RFC4787] Audet, F., Ed. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, DOI 10.17487/RFC4787, January 2007, <http://www.rfc-editor.org/info/rfc4787>.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <http://www.rfc-editor.org/info/rfc6333>.

[RFC7659] Perreault, S., Tsou, T., Sivakumar, S., and T. Taylor, "Definitions of Managed Objects for Network Address Translators (NATs)", RFC 7659, DOI 10.17487/RFC7659, October 2015, <http://www.rfc-editor.org/info/rfc7659>.

12.2. Informative References

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, DOI 10.17487/RFC2579, April 1999, <http://www.rfc-editor.org/info/rfc2579>.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <http://www.rfc-editor.org/info/rfc2629>.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, DOI 10.17487/RFC3410, December 2002, <http://www.rfc-editor.org/info/rfc3410>.

[RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, DOI 10.17487/RFC3414, December 2002, <http://www.rfc-editor.org/info/rfc3414>.

[RFC3826] Blumenthal, U., Maino, F., and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, DOI 10.17487/RFC3826, June 2004, <http://www.rfc-editor.org/info/rfc3826>.

[RFC5591] Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 5591, DOI 10.17487/RFC5591, June 2009, <http://www.rfc-editor.org/info/rfc5591>.

[RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, DOI 10.17487/RFC5592, June 2009, <http://www.rfc-editor.org/info/rfc5592>.

[RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 6353, DOI 10.17487/RFC6353, July 2011, <http://www.rfc-editor.org/info/rfc6353>.

Authors' Addresses

Yu Fu
CNNIC
No.4 South 4th Street, Zhongguancun
Hai-Dian District, Beijing, 100190
P.R. China

Email: fuyu@cnnic.cn

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
P.R. China

Email: jiangsheng@huawei.com

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

Email: knight.dongjiang@gmail.com

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

Email: flashfoxmx@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 10, 2013

O. Troan
W. Dec
Cisco Systems
X. Li
C. Bao
CERNET Center/Tsinghua University
S. Matsushima
SoftBank Telecom
T. Murakami
IP Infusion

February 06, 2013

Mapping of Address and Port with Encapsulation (MAP)
draft-ietf-softwire-map-04

Abstract

This document describes a mechanism for transporting IPv4 packets across an IPv6 network using IP encapsulation, and a generic mechanism for mapping between IPv6 addresses and IPv4 addresses and transport layer ports.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 10, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. Conventions 4
3. Terminology 4
4. Architecture 5
5. Mapping Algorithm 6
5.1. Port mapping algorithm 8
5.2. Basic mapping rule (BMR) 9
5.3. Forwarding mapping rule (FMR) 11
5.4. Destinations outside the MAP domain 11
6. The IPv6 Interface Identifier 11
7. MAP Configuration 12
7.1. MAP CE 12
7.2. MAP BR 13
7.3. Backwards compatibility 13
7.4. Address Independence 13
8. Forwarding Considerations 14
8.1. Receiving rules 14
8.2. MAP BR 14
9. ICMP 15
10. Fragmentation and Path MTU Discovery 15
10.1. Fragmentation in the MAP domain 15
10.2. Receiving IPv4 Fragments on the MAP domain borders . . . 16
10.3. Sending IPv4 fragments to the outside 16
11. NAT44 Considerations 17
12. IANA Considerations 17
13. Security Considerations 17
14. Contributors 18
15. Acknowledgements 19
16. References 19
16.1. Normative References 19
16.2. Informative References 20
Appendix A. Examples 22
Appendix B. Alternate description of the Port mapping algorithm . 26
B.1. Bit Representation of the Algorithm 27
B.2. GMA examples 27
Authors' Addresses 28

1. Introduction

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

There are numerous variations of this idea, described in 6over4 [RFC2529], 6to4 [RFC3056], ISATAP [RFC5214], and 6rd [RFC5969].

The commonalities of all these IPv6 over IPv4 mechanisms are:

- Automatically provisions an IPv6 address for a host or an IPv6 prefix for a site
- Algorithmic or implicit address resolution of tunnel end point addresses. Given an IPv6 destination address, an IPv4 tunnel endpoint address can be calculated.
- Embedding of an IPv4 address or part thereof into an IPv6 address.

In phases of IPv4 to IPv6 migration, IPv6 only networks will be common, while there will still be a need for residual IPv4 deployment. This document describes a generic mapping of IPv4 to IPv6, and a mechanism for encapsulating IPv4 over IPv6.

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

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

The mapping scheme described here supports encapsulation of IPv4 packets in IPv6 in both mesh and hub and spoke topologies, including address mappings with full independence between IPv6 and IPv4 addresses.

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

The A+P (Address and Port) architecture of sharing an IPv4 address by distributing the port space is described in [RFC6346]. Specifically section 4 of [RFC6346] covers stateless mapping. The corresponding

stateful solution DS-lite is described in [RFC6333]. The motivation for the work is described in [I-D.ietf-softwire-stateless-4v6-motivation].

A companion document defines a DHCPv6 option for provisioning of MAP [I-D.ietf-softwire-map-dhcp]. Other means of provisioning is possible. Deployment considerations are described in [I-D.mdt-softwire-map-deployment].

MAP relies on IPv6 and is designed to deliver production-quality dual-stack service while allowing IPv4 to be phased out within the SP network. The phasing out of IPv4 within the SP network is independent of whether the end user disables IPv4 service or not. Further, "Greenfield"; IPv6-only networks may use MAP in order to deliver IPv4 to sites via the IPv6 network.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Terminology

MAP domain:	One or more MAP CEs and BRs connected to the same virtual link. A service provider may deploy a single MAP domain, or may utilize multiple MAP domains.
MAP Rule	A set of parameters describing the mapping between an IPv4 prefix, IPv4 address or shared IPv4 address and an IPv6 prefix or address. Each domain uses a different mapping rule set.
MAP node	A device that implements MAP.
MAP Border Relay (BR):	A MAP enabled router managed by the service provider at the edge of a MAP domain. A Border Relay router has at least an IPv6-enabled interface and an IPv4 interface connected to the native IPv4 network. A MAP BR may also be referred to simply as a "BR" within the context of MAP.
MAP Customer Edge (CE):	A device functioning as a Customer Edge router in a MAP deployment. A typical MAP CE adopting MAP rules will serve a residential site with one WAN side interface, and one or more LAN side interfaces. A MAP CE may also be referred to simply as a "CE" within the context of MAP.

Port-set: The separate part of the transport layer port space; denoted as a port-set.

Port-set ID (PSID): Algorithmically identifies a set of ports exclusively assigned to a CE.

Shared IPv4 address: An IPv4 address that is shared among multiple CEs. Only ports that belong to the assigned port-set can be used for communication. Also known as a Port-Restricted IPv4 address.

End-user IPv6 prefix: The IPv6 prefix assigned to an End-user CE by other means than MAP itself. E.g. Provisioned using DHCPv6 PD [RFC3633], assigned via SLAAC [RFC4862], or configured manually. It is unique for each CE.

MAP IPv6 address: The IPv6 address used to reach the MAP function of a CE from other CEs and from BRs.

Rule IPv6 prefix: An IPv6 prefix assigned by a Service Provider for a mapping rule.

Rule IPv4 prefix: An IPv4 prefix assigned by a Service Provider for a mapping rule.

Embedded Address (EA) bits: The IPv4 EA-bits in the IPv6 address identify an IPv4 prefix/address (or part thereof) or a shared IPv4 address (or part thereof) and a port-set identifier.

4. Architecture

In accordance with the requirements stated above, the MAP mechanism can operate with shared IPv4 addresses, full IPv4 addresses or IPv4 prefixes. Operation with shared IPv4 addresses is described now, and the differences for full IPv4 addresses and prefixes are described below.

The MAP mechanism uses existing standard building blocks. The existing NAPT on the CE is used with additional support for restricting transport protocol ports, ICMP identifiers and fragment identifiers to the configured port set. For packets outbound from the private IPv4 network, the CE NAPT MUST translate transport identifiers (e.g. TCP and UDP port numbers) so that they fall within the CE’s assigned port-range.

The NAPT MUST in turn be connected to a MAP aware forwarding function, that does encapsulation/ decapsulation of IPv4 packets in IPv6. MAP supports the encapsulation mode specified in [RFC2473]. In addition MAP specifies an algorithm to do "address resolution" from an IPv4 address and port to an IPv6 address. This algorithmic mapping is specified in Section 5.

The MAP architecture described here, restricts the use of the shared IPv4 address to only be used as the global address (outside) of the NAPT [RFC2663] running on the CE. A shared IPv4 address MUST NOT be used to identify an interface. While it is theoretically possible to make host stacks and applications port-aware, that is considered too drastic a change to the IP model [RFC6250].

For full IPv4 addresses and IPv4 prefixes, the architecture just described applies with two differences. First, a full IPv4 address or IPv4 prefix can be used as it is today, e.g., for identifying an interface or as a DHCP pool, respectively. Secondly, the NAPT is not required to restrict the ports used on outgoing packets.

This architecture is illustrated in Figure 1.

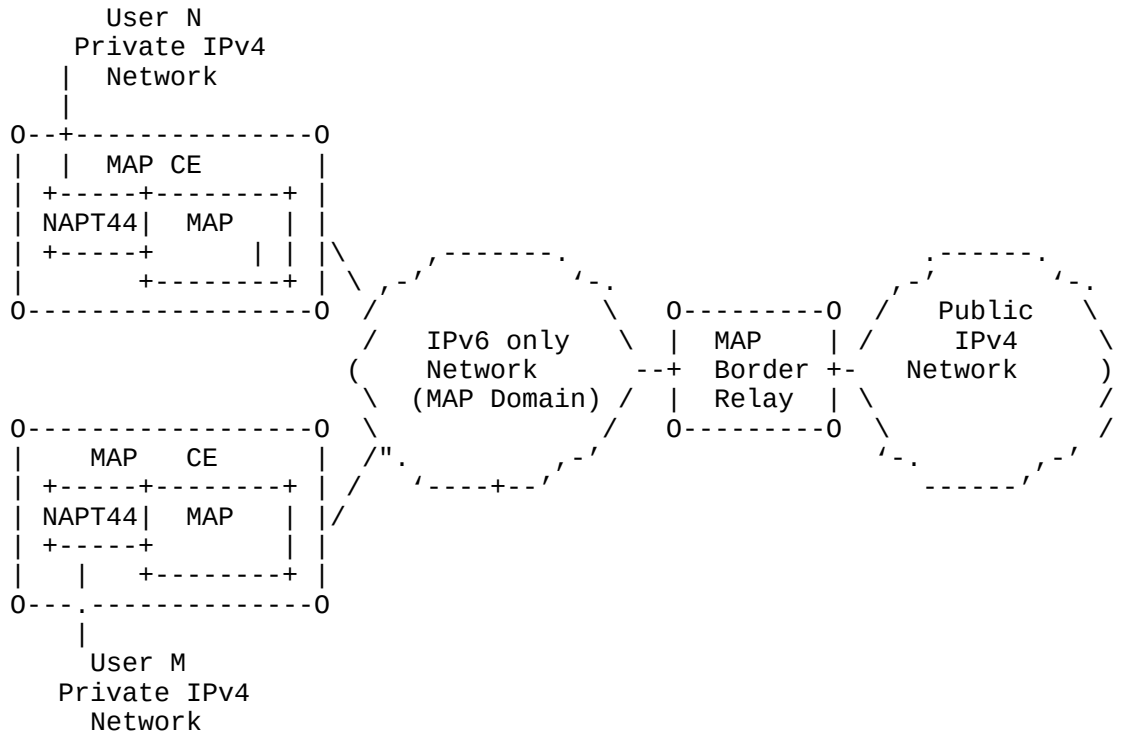


Figure 1: Network Topology

The MAP BR is responsible for connecting external IPv4 networks to the IPv4 nodes in one or more MAP domains.

5. Mapping Algorithm

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

Mapping rules are used differently depending on their function. Every MAP node must be provisioned with a Basic mapping rule. This is used by the node to configure its IPv4 address, IPv4 prefix or shared IPv4 address. This same basic rule can also be used for forwarding, where an IPv4 destination address and optionally a destination port is mapped into an IPv6 address. Additional mapping rules are specified to allow for multiple different IPv4 sub-nets to exist within the domain and optimize forwarding between them.

Traffic outside of the domain (i.e. When the destination IPv4 address does not match (using longest matching prefix) any Rule IPv4 prefix in the Rules database) is forwarded to the BR.

There are two types of mapping rules:

1. Basic Mapping Rule (BMR) - mandatory, used for IPv4 prefix, address or port set assignment. There can only be one Basic Mapping Rule per End-user IPv6 prefix. The Basic Mapping Rule is used to configure the MAP IPv6 address or prefix.

2. Forwarding Mapping Rule (FMR) - optional, used for forwarding. The Basic Mapping Rule is also a Forwarding Mapping Rule. Each Forwarding Mapping Rule will result in an entry in the Rules table for the Rule IPv4 prefix.

Both mapping rules share the same parameters:

- Rule IPv6 prefix (including prefix length)

- Rule IPv4 prefix (including prefix length)

- Rule EA-bits length (in bits)

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

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

Port-aware IPv4 entries in the Rules table are installed for all the Forwarding Mapping Rules and an IPv4 default route to the MAP BR.

Forwarding rules are used to allow direct communication between MAP CEs, known as mesh mode. In hub and spoke mode, there are no forwarding rules, all traffic MUST be forwarded directly to the BR.

5.1. Port mapping algorithm

The port mapping algorithm is used in domains whose rules allow IPv4 address sharing.

The simplest way to represent a port range is using a notation similar to CIDR [RFC4632]. For example the first 256 ports are represented as port prefix 0.0/8. The last 256 ports as 255.0/8. In hexadecimal, 0x0000/8 (PSID = 0) and 0xFF00/8 (PSID = 0xFF).

To minimise dependencies between the End-user IPv6 prefix and the resulting port set, a PSID of 0, would, in the naive representation assign the system ports [I-D.ietf-tsvwg-iana-ports] to the user. Instead using an infix representation, and requiring that the first bit field (A) is greater than 0, the well known ports are excluded.

This algorithm allocates ports to a given CE as a series of contiguous ranges.

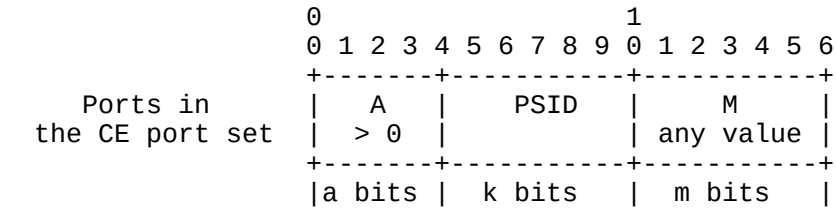


Figure 2: PSID

A For a > 0, A MUST be larger than 0. This ensures that the algorithm excludes the system ports.

a-bits The number of offset bits. The default Offset bits (a) are: 4. To simplify the port mapping algorithm the defaults are chosen so that the PSID field starts on a nibble boundary and the excluded port range (0-1023) is extended to 0-4095.

PSID The Port Set Identifier. Different Port-Set Identifiers (PSID) MUST have non-overlapping port-sets.

k-bits The length in bits of the PSID field. The sharing ratio is k^2. The number of ports assigned to the user is 2^(16-k) - 2^m (excluded ports)

M The contiguous ports.

m bits The size contiguous ports. The number of contiguous ports is given by 2^m.

This algorithm allocates ports to a given CE as a series of contiguous ranges.

5.2. Basic mapping rule (BMR)

The Basic Mapping Rule is mandatory, used by the CE to provision itself with an IPv4 prefix, IPv4 address or shared IPv4 address.

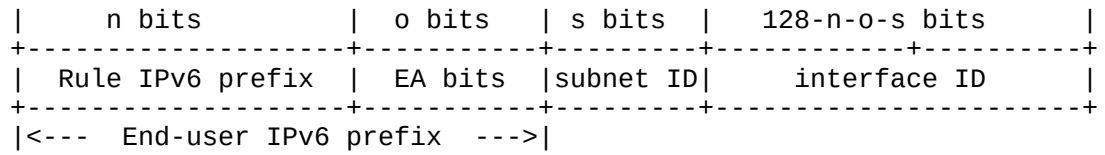


Figure 3: IPv6 address format

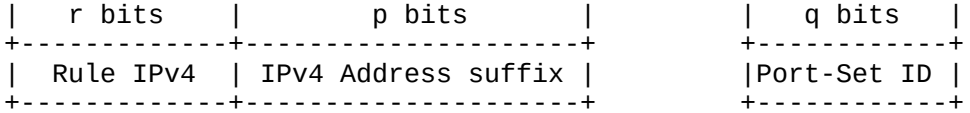
The Rule IPv6 prefix is the part of the End-user IPv6 prefix that is common among all CEs using the same Basic Mapping Rule within the MAP domain. The EA bits encode the CE specific IPv4 address and port information. The EA bits, which are unique for a given Rule IPv6 prefix, can contain a full or part of an IPv4 address and, in the shared IPv4 address case, a Port-Set Identifier (PSID). An EA-bit length of 0 signifies that all relevant MAP IPv4 addressing information is passed directly in the BMR rule, and not derived from the End-user IPv6 prefix.

The MAP IPv6 address is created by concatenating the End-user IPv6 prefix with the MAP subnet-id (if the End-user IPv6 prefix is shorter than 64 bits) and the interface-id as specified in Section 6.

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

The MAP IPv6 is created by combining the End-User IPv6 prefix with the all zeros subnet-id and the MAP IPv6 interface identifier.

Shared IPv4 address:



```
|            32 bits                |
```

Figure 4: Shared IPv4 address

Complete IPv4 address:

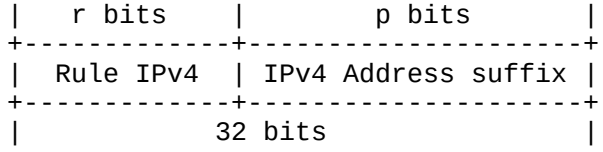


Figure 5: Complete IPv4 address

IPv4 prefix:

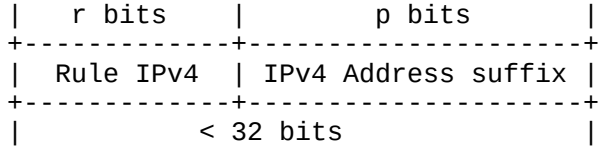


Figure 6: IPv4 prefix

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

The offset of the EA bits field in the IPv6 address is equal to the BMR Rule IPv6 prefix length. The length of the EA bits field (o) is given by the BMR Rule EA-bits length, and can be between 0 and 48. The sum of the Rule IPv6 Prefix length and the Rule EA-bits length MUST be less or equal than the End-user IPv6 prefix length.

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

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

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

The length of r MAY be 32, with no part of the IPv4 address embedded in the EA bits. This results in a mapping with no dependence between the IPv4 address and the IPv6 address. In addition the length of o MAY be zero (no EA bits embedded in the End-User IPv6 prefix), meaning that also the PSID is provisioned using e.g. the DHCP option.

See Appendix A for an example of the Basic Mapping Rule.

5.3. Forwarding mapping rule (FMR)

The Forwarding Mapping Rule is optional, and used in mesh mode to merit direct CE to CE connectivity.

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

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

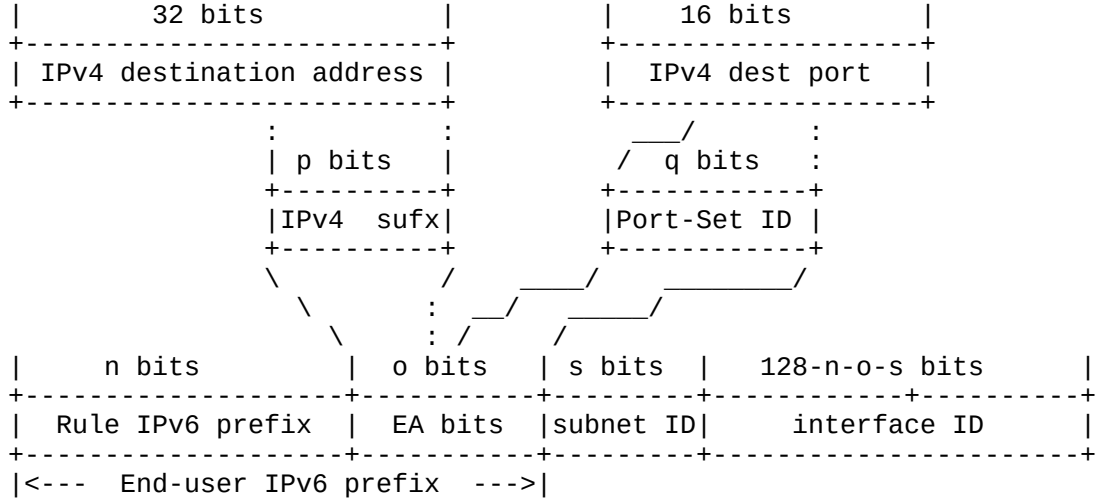


Figure 7: Deriving of MAP IPv6 address

See Appendix A for an example of the Forwarding Mapping Rule.

5.4. Destinations outside the MAP domain

To reach IPv4 destinations outside of the MAP domain, traffic is sent to the configured address of the MAP BR. On the CE, the default can be represented as a point to point IPv4 over IPv6 tunnel [RFC2473] to the BR.

6. The IPv6 Interface Identifier

The Interface identifier format of a MAP node is described below.

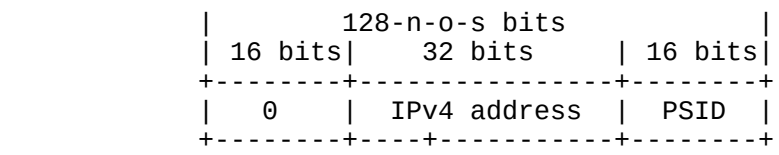


Figure 8

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

If the End-user IPv6 prefix length is larger than 64, the most significant parts of the interface identifier is overwritten by the prefix.

7. MAP Configuration

For a given MAP domain, the BR and CE MUST be configured with the following MAP elements. The configured values for these elements are identical for all CEs and BRs within a given MAP domain.

- The End-User IPv6 prefix (Part of the normal IPv6 provisioning).
- The Basic Mapping Rule and optionally the Forwarding Mapping Rules, including the Rule IPv6 prefix, Rule IPv4 prefix, and Length of EA bits
- The IPv6 address of the MAP BR.
- Hub and spoke mode or Mesh mode. (If all traffic should be sent to the BR, or if direct CE to CE traffic should be supported).

7.1. MAP CE

The MAP elements are set to values that are the same across all CEs within a MAP domain. The values may be configured in a variety of manners, including provisioning methods such as the Broadband Forum’s

"TR-69" Residential Gateway management interface, an XML-based object retrieved after IPv6 connectivity is established, or manual configuration by an administrator. This document describes how to configure the necessary parameters via a single IPv6 DHCP option. A CE that allows IPv6 configuration by DHCP SHOULD implement this option. Other configuration and management methods may use the format described by this option for consistency and convenience of implementation on CEs that support multiple configuration methods.

The only remaining provisioning information the CE requires in order to calculate the MAP IPv4 address and enable IPv4 connectivity is the IPv6 prefix for the CE. The End-user IPv6 prefix is configured as part of obtaining IPv6 Internet access.

A single MAP CE MAY be connected to more than one MAP domain, just as any router may have more than one IPv4-enabled service provider facing interface and more than one set of associated addresses assigned by DHCP. Each domain a given CE operates within would require its own set of MAP configuration elements and would generate its own IPv4 address.

The MAP DHCP option is specified in [I-D.ietf-softwire-map-dhcp].

7.2. MAP BR

The MAP BR MUST be configured with the same MAP elements as the MAP CEs operating within the same domain.

For increased reliability and load balancing, the BR IPv6 address MAY be an anycast address shared across a given MAP domain. As MAP is stateless, any BR may be used at any time. If the BR IPv6 address is anycast the relay MUST use this anycast IPv6 address as the source address in packets relayed to CEs.

Since MAP uses provider address space, no specific routes need to be advertised externally for MAP to operate, neither in IPv6 nor IPv4 BGP. However, if anycast is used for the MAP IPv6 relays, the anycast addresses must be advertised in the service provider’s IGP.

7.3. Backwards compatibility

A MAP-E CE provisioned with only the IPv6 address of the BR, and with no IPv4 address and port range configured by other means, MUST disable its NAT44 functionality. This characteristic makes a MAP CE compatible with DS-Lite [RFC6333] AFTRs, whose addresses are configured as the MAP BR.

7.4. Address Independence

The MAP solution supports use and configuration of domains in so called 1:1 mode (meaning 1 mapping rule set per CE), which allows complete independence between the IPv6 prefix assigned to the CE and

the IPv4 address and/or port-range it uses. This is achieved in all cases when the EA-bit length is set to 0.

The constraint imposed is that each such MAP domain be composed of just 1 MAP CE which has a predetermined IPv6 prefix, i.e. The BR would be configured with a rule-set per CPE, where the FMR would uniquely describe the IPv6 prefix of a given CE. Each CE would have a distinct BMR, that would fully describe that CE's IPv4 address, and PSID if any.

8. Forwarding Considerations

Figure 1 depicts the overall MAP architecture with IPv4 users (N and M) networks connected to a routed IPv6 network.

MAP supports Encapsulation mode as specified in [RFC2473].

For a shared IPv4 address, a MAP CE forwarding IPv4 packets from the LAN performs NAT44 functions first and creates appropriate NAT44 bindings. The resulting IPv4 packets MUST contain the source IPv4 address and source transport identifiers defined by MAP. The resulting IPv4 packet is forwarded to the CE's MAP forwarding function. The IPv6 source and destination addresses MUST then be derived as per Section 5 of this draft.

A MAP CE receiving an IPv6 packet to its MAP IPv6 address sends this packet to the CE's MAP function. All other IPv6 traffic is forwarded as per the CE's IPv6 routing rules. The resulting IPv4 packet is then forwarded to the CE's NAT44 function where the destination port number MUST be checked against the stateful port mapping session table and the destination port number MUST be mapped to its original value.

8.1. Receiving rules

The CE SHOULD check that MAP received packets' transport-layer destination port number is in the range configured by MAP for the CE and the CE SHOULD drop any non conforming packet and respond with an ICMPv6 "Address Unreachable" (Type 1, Code 3).

8.2. MAP BR

A MAP BR receiving IPv6 packets selects a best matching MAP domain rule based on a longest address match of the packets' source address against the BR's configured MAP BMR prefix(es), as well as a match of the packet destination address against the configured BR IPv6 address or FMR prefix(es). The selected MAP rule allows the BR to determine the EA-bits from the source IPv6 address. The BR MUST perform a validation of the consistency of the source IPv6 address and source

port number for the packet using BMR. If the packets source port number is found to be outside the range allowed for this CE and the BMR, the BR MUST drop the packet and respond with an ICMPv6 "Destination Unreachable, Source address failed ingress/egress policy" (Type 1, Code 5).

9. ICMP

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

If a MAP CE receives an ICMP message having ICMP identifier field in ICMP header, NAT44 in the MAP CE must rewrite this field to a specific value assigned from the port-set. BR and other CEs must handle this field similar to the port number in the TCP/UDP header upon receiving the ICMP message with ICMP identifier field.

If a MAP node receives an ICMP error message without the ICMP identifier field for errors that is detected inside a IPv6 tunnel, a node should relay the ICMP error message to the original source. This behavior should be implemented conforming to the section 8 of [RFC2473].

10. Fragmentation and Path MTU Discovery

Due to the different sizes of the IPv4 and IPv6 header, handling the maximum packet size is relevant for the operation of any system connecting the two address families. There are three mechanisms to handle this issue: Path MTU discovery (PMTUD), fragmentation, and transport-layer negotiation such as the TCP Maximum Segment Size (MSS) option [RFC0897]. MAP uses all three mechanisms to deal with different cases.

10.1. Fragmentation in the MAP domain

Encapsulating an IPv4 packet to carry it across the MAP domain will increase its size (40 bytes). It is strongly recommended that the MTU in the MAP domain is well managed and that the IPv6 MTU on the CE WAN side interface is set so that no fragmentation occurs within the boundary of the MAP domain.

Fragmentation on MAP domain entry is described in section 7.2 of [RFC2473]

The use of an anycast source address could lead to any ICMP error message generated on the path being sent to a different BR. Therefore, using dynamic tunnel MTU Section 6.7 of [RFC2473] is subject to IPv6 Path MTU black-holes. A MAP BR SHOULD NOT by default use Path MTU discovery across the MAP domain.

Multiple BRs using the same anycast source address could send fragmented packets to the same CE at the same time. If the fragmented packets from different BRs happen to use the same fragment ID, incorrect reassembly might occur. See [RFC4459] for an analysis of the problem. Section 3.4 suggests solving the problem by fragmenting the inner packet.

10.2. Receiving IPv4 Fragments on the MAP domain borders

Forwarding of an IPv4 packet received from the outside of the MAP domain requires the IPv4 destination address and the transport protocol destination port. The transport protocol information is only available in the first fragment received. As described in section 5.3.3 of [RFC6346] a MAP node receiving an IPv4 fragmented packet from outside has to reassemble the packet before sending the packet onto the MAP link. If the first packet received contains the transport protocol information, it is possible to optimize this behavior by using a cache and forwarding the fragments unchanged. A description of this algorithm is outside the scope of this document.

10.3. Sending IPv4 fragments to the outside

If two IPv4 host behind two different MAP CE’s with the same IPv4

address sends fragments to an IPv4 destination host outside the domain. Those hosts may use the same IPv4 fragmentation identifier, resulting in incorrect reassembly of the fragments at the destination host. Given that the IPv4 fragmentation identifier is a 16 bit field, it could be used similarly to port ranges. A MAP CE SHOULD rewrite the IPv4 fragmentation identifier to be within its allocated port set.

11. NAT44 Considerations

The NAT44 implemented in the MAP CE SHOULD conform with the behavior and best current practice documented in [RFC4787], [RFC5508], and [RFC5382]. In MAP address sharing mode (determined by the MAP domain /rule configuration parameters) the operation of the NAT44 MUST be restricted to the available port numbers derived via the basic mapping rule.

12. IANA Considerations

This specification does not require any IANA actions.

13. Security Considerations

Spoofing attacks: With consistency checks between IPv4 and IPv6 sources that are performed on IPv4/IPv6 packets received by MAP nodes, MAP does not introduce any new opportunity for spoofing attacks that would not already exist in IPv6.

Denial-of-service attacks: In MAP domains where IPv4 addresses are shared, the fact that IPv4 datagram reassembly may be necessary introduces an opportunity for DOS attacks. This is inherent to address sharing, and is common with other address sharing approaches such as DS-Lite and NAT64/DNS64. The best protection against such attacks is to accelerate IPv6 deployment, so that, where MAP is supported, it is less and less used.

Routing-loop attacks: This attack may exist in some automatic

tunneling scenarios are documented in [RFC6324]. They cannot exist with MAP because each BRs checks that the IPv6 source address of a received IPv6 packet is a CE address based on Forwarding Mapping Rule.

Attacks facilitated by restricted port set: From hosts that are not subject to ingress filtering of [RFC2827], some attacks are possible by an attacker injecting spoofed packets during ongoing transport connections ([RFC4953], [RFC5961], [RFC6056]. The attacks depend on guessing which ports are currently used by target hosts, and using an unrestricted port set is preferable, i.e. Using native IPv6 connections that are not subject to MAP port range restrictions. To minimize this type of attacks when using a restricted port set, the MAP CE's NAT44 filtering behavior SHOULD be "Address-Dependent Filtering". Furthermore, the MAP CEs SHOULD use a DNS transport proxy function to handle DNS traffic, and source such traffic from IPv6 interfaces not assigned to MAP. Practicalities of these methods are discussed in Section 5.9 of [I-D.dec-stateless-4v6].

[RFC6269] outlines general issues with IPv4 address sharing.

14. Contributors

This document is the result of the IETF Softwire MAP design team effort and numerous previous individual contributions in this area:

Chongfeng Xie (China Telecom)
Room 708, No.118, Xizhimennei Street Beijing 100035 CN
Phone: +86-10-58552116
Email: xiechf@ctbri.com.cn

Qiong Sun (China Telecom)
Room 708, No.118, Xizhimennei Street Beijing 100035 CN
Phone: +86-10-58552936
Email: sunqiong@ctbri.com.cn

Gang Chen (China Mobile)
53A,Xibianmennei Ave. Beijing 100053 P.R.China
Email: chengang@chinamobile.com

Yu Zhai
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN
Email: jacky.zhai@gmail.com

Wentao Shang (CERNET Center/Tsinghua University)
Room 225, Main Building, Tsinghua University Beijing 100084
CN
Email: wentaoshang@gmail.com

Guoliang Han (CERNET Center/Tsinghua University)
Room 225, Main Building, Tsinghua University Beijing 100084
CN
Email: bupthgl@gmail.com

Rajiv Asati (Cisco Systems)
7025-6 Kit Creek Road Research Triangle Park NC 27709 USA
Email: rajiva@cisco.com

15. Acknowledgements

This document is based on the ideas of many, including Masakazu Asama, Mohamed Boucadair, Gang Chen, Maoke Chen, Wojciech Dec, Xiaohong Deng, Jouni Korhonen, Tomasz Mrugalski, Jacni Qin, Chunfa Sun, Qiong Sun, and Leaf Yeh. The authors want in particular to recognize Remi Despres, who has tirelessly worked on generalized mechanisms for stateless address mapping.

The authors would like to thank Guillaume Gottard, Dan Wing, Jan Zorz, Necj Scoberne, Tina Tsou and especially Tom Taylor for the thorough review and comments of this document.

16. References

16.1. Normative References

[I-D.ietf-softwire-map-dhcp]
Mrugalski, T., Troan, O., Bao, C., Dec, W., and L. Yeh, "DHCPv6 Options for Mapping of Address and Port", draft-ietf-softwire-map-dhcp-01 (work in progress), August 2012.

[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.

16.2. Informative References

[I-D.dec-stateless-4v6]
Dec, W., Asati, R., and H. Deng, "Stateless 4Via6 Address Sharing", draft-dec-stateless-4v6-04 (work in progress), October 2011.

[I-D.ietf-softwire-stateless-4v6-motivation]
Boucadair, M., Matsushima, S., Lee, Y., Bonness, O., Borges, I., and G. Chen, "Motivations for Carrier-side Stateless IPv4 over IPv6 Migration Solutions", draft-ietf-softwire-stateless-4v6-motivation-05 (work in progress), November 2012.

[I-D.ietf-tsvwg-iana-ports]
Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", draft-ietf-tsvwg-iana-ports-10 (work in progress), February 2011.

[RFC0897] Postel, J., "Domain name system implementation schedule", RFC 897, February 1984.

[RFC1933] Gilligan, R. and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 1933, April 1996.

[RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, March 1999.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

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

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

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, April 2006.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

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

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

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

[RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.

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

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

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

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

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

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-

Protocol Port Randomization", BCP 156, RFC 6056, January 2011.

[RFC6250] Thaler, D., "Evolution of the IP Model", RFC 6250, May 2011.

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

[RFC6324] Nakibly, G. and F. Templin, "Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", RFC 6324, August 2011.

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

[RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.

Appendix A. Examples

Example 1 - BMR

Given the MAP domain information and an IPv6 address of an endpoint:

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

A MAP node (CE or BR) can via the BMR, or equivalent FMR, determine the IPv4 address and port-set as shown below:

EA bits offset: 40
IPv4 suffix bits (p) Length of IPv4 address (32) - IPv4 prefix length (24) = 8
IPv4 address 192.0.2.18 (0xc0000212)
PSID start: 40 + p = 40 + 8 = 48
PSID length: o - p = (56 - 40) - 8 = 8
PSID: 0x34

Port-set-1: 4928, 4929, 4930, 4931, 4932, 4933, 4934, 4935, 4936, 4937, 4938, 4939, 4940, 4941, 4942, 4943
Port-set-2: 9024, 9025, 9026, 9027, 9028, 9029, 9030, 9031, 9032, 9033, 9034, 9035, 9036, 9037, 9038, 9039
... ...
Port-set-15 62272, 62273, 62274, 62275, 62276, 62277, 62278, 62279, 62280, 62281, 62282, 62283, 62284, 62285, 62286, 62287

The BMR information allows a MAP CE also to determine (complete) its IPv6 address within the indicated IPv6 prefix.

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

Example 2:

Another example can be made of a hypothetical MAP BR, configured with the following FMR when receiving a packet with the following characteristics:

IPv4 source address: 1.2.3.4 (0x01020304)
IPv4 source port: 80
IPv4 destination address: 192.0.2.18 (0xc0000212)
IPv4 destination port: 9030

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

IPv6 address of MAP BR: 2001:db8:ffff::1

The above information allows the BR to derive as follows the mapped destination IPv6 address for the corresponding MAP CE, and also the mapped source IPv6 address for the IPv4 source.

IPv4 suffix bits (p) 32 - 24 = 8 (18 (0x12))
PSID length: 8
PSID: 0x34 (9030 (0x2346))

The resulting IPv6 packet will have the following key fields:

IPv6 source address: 2001:db8:ffff::1
IPv6 destination address: 2001:db8:0012:3400:00c0:0002:1200:3400
IPv6 source Port: 80
IPv6 destination Port: 9030

Example 3 - FMR:

An IPv4 host behind the MAP CE (addressed as per the previous examples) corresponding with IPv4 host 1.2.3.4 will have its packets converted into IPv6 using the IPv6 address of the BR configured on the MAP CE as follows:

IPv6 address of BR used by MAP CE: 2001:db8:ffff::1
IPv4 source address (post NAT44 if present) 192.0.2.18
IPv4 destination address: 1.2.3.4
IPv4 source port (post NAT44 if present): 9030
IPv4 destination port: 80
IPv6 source address of MAP CE:
2001:db8:0012:3400:00c0:0002:1200:3400
IPv6 destination address: 2001:db8:ffff::1

Example 4 - 1:1 Rule with no address sharing

IPv6 prefix assigned to the end user: 2001:db8:0012:3400::/56
Basic Mapping Rule: {2001:db8:0012:3400::/56 (Rule IPv6 prefix),
192.0.2.1/32 (Rule IPv4 prefix), 0 (Rule EA-bits length)}
PSID length: 0 (Sharing ratio is 1)
PSID offset: n/a

A MAP node (CE or BR) can via the BMR or equivalent FMR, determine the IPv4 address and port-set as shown below:

EA bits offset: 0
IPv4 suffix bits (p) Length of IPv4 address (32) - IPv4 prefix length (32) = 0
IPv4 address 192.0.2.1 (0xc0000201)
PSID start: 0
PSID length: 0
PSID: null

The BMR information allows a MAP CE also to determine (complete) its full IPv6 address by combining the IPv6 prefix with the MAP interface identifier (that embeds the IPv4 address).

IPv6 address of MAP CE: 2001:db8:0012:3400:00c0:0002:0100:0000

Example 5 - 1:1 Rule with address sharing (sharing ratio 256)

IPv6 prefix assigned to the end user: 2001:db8:0012:3400::/56
Basic Mapping Rule: {2001:db8:0012:3400::/56 (Rule IPv6 prefix), 192.0.2.1/32 (Rule IPv4 prefix), 0 (Rule EA-bits length)}
PSID length: (16 - (32 - 24) = 8. (Sharing ratio of 256)
PSID offset: 4

A MAP node (CE or BR) can via the BMR or equivalent FMR determine the IPv4 address and port-set as shown below:

EA bits offset: 0
IPv4 suffix bits (p) Length of IPv4 address (32) - IPv4 prefix length (32) = 0
IPv4 address 192.0.2.1 (0xc0000201)
PSID start: 0
PSID length: 8
PSID: 0x34

Port-set-1: 4928, 4929, 4930, 4931, 4932, 4933, 4934, 4935, 4936, 4937, 4938, 4939, 4940, 4941, 4942, 4943
Port-set-2: 9024, 9025, 9026, 9027, 9028, 9029, 9030, 9031, 9032, 9033, 9034, 9035, 9036, 9037, 9038, 9039
... ...
Port-set-15 62272, 62273, 62274, 62275, 62276, 62277, 62278, 62279, 62280, 62281, 62282, 62283, 62284, 62285, 62286, 62287

The BMR information allows a MAP CE also to determine (complete) its full IPv6 address by combining the IPv6 prefix with the MAP interface identifier (that embeds the IPv4 address and PSID).

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

Note that the IPv4 address and PSID is not derived from the IPv6 prefix assigned to the CE.

Appendix B. Alternate description of the Port mapping algorithm

The port mapping algorithm is used in domains whose rules allow IPv4 address sharing. Different Port-Set Identifiers (PSID) MUST have non-overlapping port-sets. The two extreme cases are: (1) the port numbers are not contiguous for each PSID, but uniformly distributed across the port range (0-65535); (2) the port numbers are contiguous in a single range for each PSID. The port mapping algorithm proposed here is called the Generalized Modulus Algorithm (GMA) and supports both these cases.

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

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

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

Where:

* PSID: K = 0 to R - 1

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

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

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

K = (floor(P/M)) % R

Where:

* % is the modulus operator

* floor(arg) is a function that returns the largest integer not greater than arg.

B.1. Bit Representation of the Algorithm

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

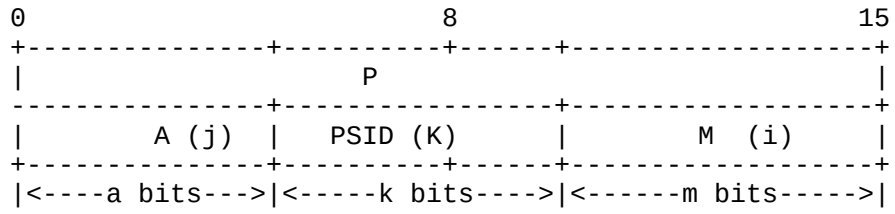


Figure 9: Bit representation

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

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

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

B.2. GMA examples

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

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

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

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

Authors' Addresses

Ole Troan
Cisco Systems
Philip Pedersens vei 1
Lysaker 1366
Norway

Email: ot@cisco.com

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

Email: wdec@cisco.com

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

Email: xing@cernet.edu.cn

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

Email: congxiao@cernet.edu.cn

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

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

Tetsuya Murakami
IP Infusion
1188 East Arques Avenue
Sunnyvale
USA

Email: tetsuya@ipinfusion.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2015

O. Troan, Ed.
W. Dec
Cisco Systems
X. Li
C. Bao
CERNET Center/Tsinghua University
S. Matsushima
SoftBank Telecom
T. Murakami
IP Infusion
T. Taylor, Ed.
Huawei Technologies
March 09, 2015

Mapping of Address and Port with Encapsulation (MAP)
draft-ietf-softwire-map-13

Abstract

This document describes a mechanism for transporting IPv4 packets across an IPv6 network using IP encapsulation, and a generic mechanism for mapping between IPv6 addresses and IPv4 addresses and transport layer ports.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 10, 2015.

Copyright Notice

Copyright (c) 2015 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. Architecture 5
5. Mapping Algorithm 7
5.1. Port mapping algorithm 8
5.2. Basic mapping rule (BMR) 10
5.3. Forwarding mapping rule (FMR) 12
5.4. Destinations outside the MAP domain 13
6. The IPv6 Interface Identifier 13
7. MAP Configuration 14
7.1. MAP CE 14
7.2. MAP BR 15
8. Forwarding Considerations 15
8.1. Receiving Rules 15
8.2. ICMP 16
8.3. Fragmentation and Path MTU Discovery 17
8.3.1. Fragmentation in the MAP domain 17
8.3.2. Receiving IPv4 Fragments on the MAP domain borders . 17
8.3.3. Sending IPv4 fragments to the outside 18
9. NAT44 Considerations 18
10. IANA Considerations 18
11. Security Considerations 18
12. Contributors 19
13. Acknowledgments 20
14. References 20
14.1. Normative References 20
14.2. Informative References 21
Appendix A. Examples 23
Appendix B. A More Detailed Description of the Derivation of the Port Mapping Algorithm 27
B.1. Bit Representation of the Algorithm 29
B.2. GMA examples 30
Authors' Addresses 30

1. Introduction

Mapping of IPv4 addresses in IPv6 addresses has been described in numerous mechanisms dating back to 1995 [RFC1933]. The Automatic tunneling mechanism described in RFC1933 assigned a globally unique IPv6 address to a host by combining the host's IPv4 address with a well-known IPv6 prefix. Given an IPv6 packet with a destination address with an embedded IPv4 address, a node could automatically tunnel this packet by extracting the IPv4 tunnel end-point address from the IPv6 destination address.

There are numerous variations of this idea, described in 6over4 [RFC2529], 6to4 [RFC3056], ISATAP [RFC5214], and 6rd [RFC5969].

The commonalities of all these IPv6 over IPv4 mechanisms are:

- Automatically provisions an IPv6 address for a host or an IPv6 prefix for a site
- Algorithmic or implicit address resolution of tunnel end point addresses. Given an IPv6 destination address, an IPv4 tunnel endpoint address can be calculated.
- Embedding of an IPv4 address or part thereof into an IPv6 address.

In later phases of IPv4 to IPv6 migration, it is expected that IPv6-only networks will be common, while there will still be a need for residual IPv4 deployment. This document describes a generic mapping of IPv4 to IPv6, and a mechanism for encapsulating IPv4 over IPv6.

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

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

The mapping scheme described here supports encapsulation of IPv4 packets in IPv6 in both mesh and hub-and-spoke topologies, including address mappings with full independence between IPv6 and IPv4 addresses.

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

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

A companion document defines a DHCPv6 option for provisioning of MAP [I-D.ietf-softwire-map-dhcp]. Other means of provisioning are possible. Deployment considerations are described in [I-D.ietf-softwire-map-deployment].

MAP relies on IPv6 and is designed to deliver dual-stack service while allowing IPv4 to be phased out within the service provider's (SP) network. The phasing out of IPv4 within the SP network is independent of whether the end user disables IPv4 service or not. Further, "greenfield"; IPv6-only networks may use MAP in order to deliver IPv4 to sites via the IPv6 network.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Terminology

MAP domain:	One or more MAP CEs and BRs connected to the same virtual link. A service provider may deploy a single MAP domain, or may utilize multiple MAP domains.
MAP rule	A set of parameters describing the mapping between an IPv4 prefix, IPv4 address or shared IPv4 address and an IPv6 prefix or address. Each domain uses a different mapping rule set.
MAP node	A device that implements MAP.
MAP Border Relay (BR):	A MAP enabled router managed by the service provider at the edge of a MAP domain. A Border Relay router has at least an IPv6-enabled interface and an IPv4 interface connected to the native IPv4 network. A MAP BR may also be referred to simply as a "BR" within the context of MAP.

MAP Customer Edge (CE): A device functioning as a Customer Edge router in a MAP deployment. A typical MAP CE adopting MAP rules will serve a residential site with one WAN side interface, and one or more LAN side interfaces. A MAP CE may also be referred to simply as a "CE" within the context of MAP.

Port-set: The separate part of the transport layer port space; denoted as a port-set.

Port-set ID (PSID): Algorithmically identifies a set of ports exclusively assigned to a CE.

Shared IPv4 address: An IPv4 address that is shared among multiple CEs. Only ports that belong to the assigned port-set can be used for communication. Also known as a Port-Restricted IPv4 address.

End-user IPv6 prefix: The IPv6 prefix assigned to an End-user CE by other means than MAP itself. E.g., Provisioned using DHCPv6 PD [RFC3633], assigned via SLAAC [RFC4862], or configured manually. It is unique for each CE.

MAP IPv6 address: The IPv6 address used to reach the MAP function of a CE from other CEs and from BRs.

Rule IPv6 prefix: An IPv6 prefix assigned by a Service Provider for a mapping rule.

Rule IPv4 prefix: An IPv4 prefix assigned by a Service Provider for a mapping rule.

Embedded Address (EA) bits: The IPv4 EA-bits in the IPv6 address identify an IPv4 prefix/address (or part thereof) or a shared IPv4 address (or part thereof) and a port-set identifier.

4. Architecture

In accordance with the requirements stated above, the MAP mechanism can operate with shared IPv4 addresses, full IPv4 addresses or IPv4 prefixes. Operation with shared IPv4 addresses is described here, and the differences for full IPv4 addresses and prefixes are described below.

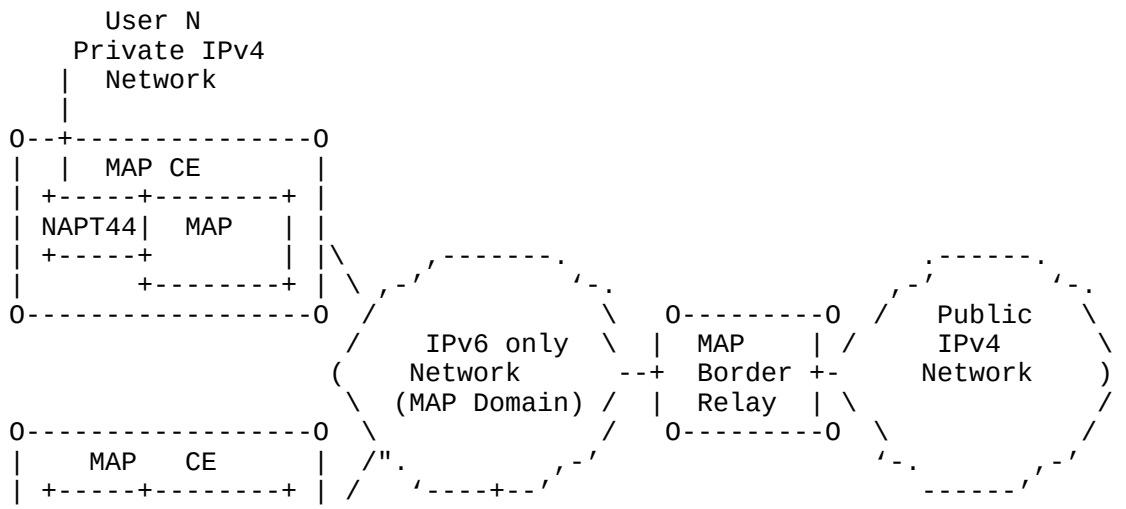
The MAP mechanism uses existing standard building blocks. The existing NAPT [RFC2663] on the CE is used with additional support for restricting transport protocol ports, ICMP identifiers and fragment identifiers to the configured port-set. For packets outbound from the private IPv4 network, the CE NAPT MUST translate transport identifiers (e.g., TCP and UDP port numbers) so that they fall within the CE's assigned port-range.

The NAPT MUST in turn be connected to a MAP-aware forwarding function, that does encapsulation / decapsulation of IPv4 packets in IPv6. MAP supports the encapsulation mode specified in [RFC2473]. In addition MAP specifies an algorithm to do "address resolution" from an IPv4 address and port to an IPv6 address. This algorithmic mapping is specified in Section 5.

The MAP architecture described here restricts the use of the shared IPv4 address to only be used as the global address (outside) of the NAPT running on the CE. A shared IPv4 address MUST NOT be used to identify an interface. While it is theoretically possible to make host stacks and applications port-aware, it would be a drastic change to the IP model [RFC6250].

For full IPv4 addresses and IPv4 prefixes, the architecture just described applies with two differences. First, a full IPv4 address or IPv4 prefix can be used as it is today, e.g., for identifying an interface or as a DHCP pool, respectively. Secondly, the NAPT is not required to restrict the ports used on outgoing packets.

This architecture is illustrated in Figure 1.



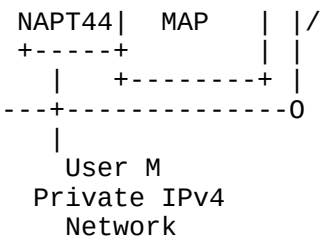


Figure 1: Network Topology

The MAP BR connects one or more MAP domains to external IPv4 networks.

5. Mapping Algorithm

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

Mapping rules are used differently depending on their function. Every MAP node must be provisioned with a Basic mapping rule. This is used by the node to configure its IPv4 address, IPv4 prefix or shared IPv4 address. This same basic rule can also be used for forwarding, where an IPv4 destination address and optionally a destination port are mapped into an IPv6 address. Additional mapping rules are specified to allow for multiple different IPv4 sub-nets to exist within the domain and optimize forwarding between them.

Traffic outside of the domain (i.e., when the destination IPv4 address does not match (using longest matching prefix) any Rule IPv4 prefix in the Rules database) is forwarded to the BR.

There are two types of mapping rules:

1. Basic Mapping Rule (BMR) - mandatory. A CE can be provisioned with multiple End-user IPv6 prefixes. There can only be one Basic Mapping Rule per End-user IPv6 prefix. However all CE’s having End-user IPv6 prefixes within (aggregated by) the same Rule IPv6 prefix may share the same Basic Mapping Rule. In combination with the End-user IPv6 prefix, the Basic Mapping Rule is used to derive the IPv4 prefix, address, or shared address and the PSID assigned to the CE.

2. Forwarding Mapping Rule (FMR) - optional, used for forwarding. The Basic Mapping Rule may also be a Forwarding Mapping Rule. Each Forwarding Mapping Rule will result in an entry in the Rules table for the Rule IPv4 prefix. Given a destination IPv4 address and port within the MAP domain, a MAP node can use the matching

FMR to derive the End-user IPv6 address of the interface through which that IPv4 destination address and port combination can be reached. In hub and spoke mode there are no FMRs.

Both mapping rules share the same parameters:

- Rule IPv6 prefix (including prefix length)
- Rule IPv4 prefix (including prefix length)
- Rule EA-bits length (in bits)

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

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

Port-restricted IPv4 routes are installed in the Rules table for all the Forwarding Mapping Rules, and a default route is installed to the MAP BR (see Section 5.4).

Forwarding Mapping Rules are used to allow direct communication between MAP CEs, known as mesh mode. In hub and spoke mode, there are no forwarding mapping rules, all traffic MUST be forwarded directly to the BR.

While an FMR is optional in the sense that a MAP CE MAY be configured with zero or more FMRs depending on the deployment, all MAP CEs MUST implement support for both rule types.

5.1. Port mapping algorithm

The port mapping algorithm is used in domains whose rules allow IPv4 address sharing.

The simplest way to represent a port range is using a notation similar to CIDR [RFC4632]. For example the first 256 ports are represented as port prefix 0.0/8. The last 256 ports as 255.0/8. In hexadecimal, 0x0000/8 (PSID = 0) and 0xFF00/8 (PSID = 0xFF). Using this technique, but wishing to avoid allocating the system ports [RFC6335] to the user, one would have to exclude the use of one or more PSIDs (e.g., PSIDs 0 to 3 in the example just given).

When the PSID is embedded in the End-user IPv6 prefix, then to minimize dependencies between the End-user IPv6 prefix and the assigned port-set, it is desirable to minimize the restrictions of possible PSID values. This is achieved by using an infix representation of the port value. Using such a representation, the well-known ports are excluded by restrictions on the value of the high-order bitfield (A) rather than the PSID.

The infix algorithm allocates ports to a given CE as a series of contiguous ranges spaced at regular intervals throughout the complete range of possible port-set values.

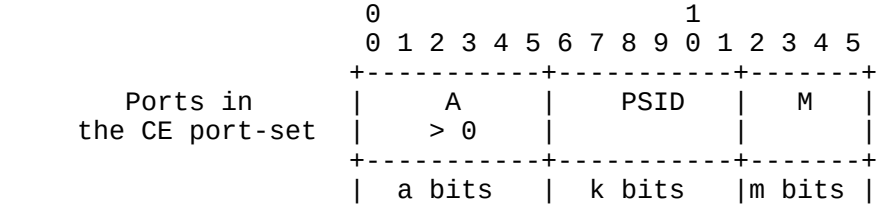


Figure 2: Structure of a port-restricted port field

a bits: The number of offset bits. 6 by default as this excludes the system ports (0-1023). To guarantee non-overlapping port sets, the offset 'a' MUST be the same for every MAP CE sharing the same address.

A: Selects the range of the port number. For 'a' > 0, A MUST be larger than 0. This ensures that the algorithm excludes the system ports. For the default value of 'a' (6), the system ports, are excluded by requiring that A be greater than 0. Smaller values of 'a' excludes a larger initial range. E.g., 'a' = 4, will exclude ports 0 - 4095. The interval between initial port numbers of successive contiguous ranges assigned to the same user is 2^(16-a).

k bits: The length in bits of the PSID field. To guarantee non-overlapping port sets, the length 'k' MUST be the same for every MAP CE sharing the same address. The sharing ratio is 2^k. The number of ports assigned to the user is 2^(16-k) - 2^m (excluded ports)

PSID: The Port-Set Identifier (PSID). Different PSID values guarantee non-overlapping port-sets thanks to the restrictions on 'a' and 'k' stated above, because the PSID always occupies the same bit positions in the port number.

m bits: The number of contiguous ports is given by 2^m.

M: Selects the specific port within a particular range specified by the concatenation of A and the PSID.

5.2. Basic mapping rule (BMR)

The Basic Mapping Rule is mandatory, used by the CE to provision itself with an IPv4 prefix, IPv4 address or shared IPv4 address. Recall from Section 5 that the BMR consists of the following parameters:

- Rule IPv6 prefix (including prefix length)
- Rule IPv4 prefix (including prefix length)
- Rule EA-bits length (in bits)

Figure 3 shows the structure of the complete MAP IPv6 address as specified in this document.

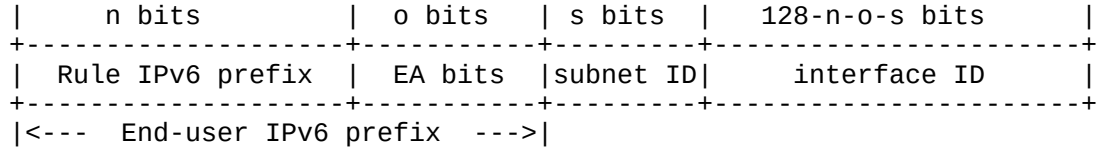


Figure 3: MAP IPv6 Address Format

The Rule IPv6 prefix (which is part of the End-user IPv6 prefix) that is common among all CEs using the same Basic Mapping Rule within the MAP domain. The EA bits encode the CE specific IPv4 address and port information. The EA bits, which are unique for a given Rule IPv6 prefix, can contain a full or part of an IPv4 address and, in the shared IPv4 address case, a Port-Set Identifier (PSID). An EA-bit length of 0 signifies that all relevant MAP IPv4 addressing information is passed directly in the BMR, and not derived from the End-user IPv6 prefix.

The MAP IPv6 address is created by concatenating the End-user IPv6 prefix with the MAP subnet identifier (if the End-user IPv6 prefix is shorter than 64 bits) and the interface identifier as specified in Section 6.

The MAP subnet identifier is defined to be the first subnet (s bits set to zero).

Define:

r = length of the IPv4 prefix given by the BMR;

o = length of the EA bit field as given by the BMR;

p = length of the IPv4 suffix contained in the EA bit field.

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

The offset of the EA bits field in the IPv6 address is equal to the BMR Rule IPv6 prefix length. The length of the EA bits field (o) is given by the BMR Rule EA-bits length, and can be between 0 and 48. A length of 48 means that the complete IPv4 address and port is embedded in the End-user IPv6 prefix (a single port is assigned). A length of 0 means that no part of the IPv4 address or port is embedded in the address. The sum of the Rule IPv6 Prefix length and the Rule EA-bits length MUST be less or equal than the End-user IPv6 prefix length.

If o + r < 32 (length of the IPv4 address in bits), then an IPv4 prefix is assigned. This case is shown in Figure 4.

IPv4 prefix:

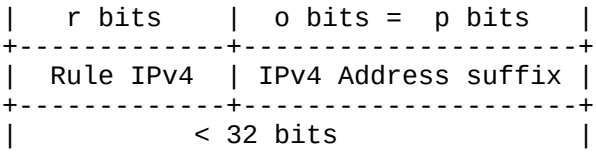


Figure 4: IPv4 prefix

If o + r is equal to 32, then a full IPv4 address is to be assigned. The address is created by concatenating the Rule IPv4 prefix and the EA-bits. This case is shown in Figure 5.

Complete IPv4 address:

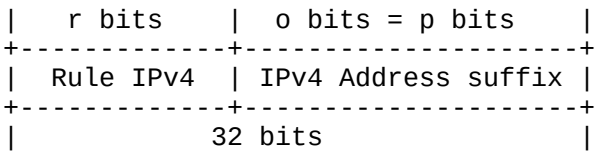


Figure 5: Complete IPv4 address

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

Shared IPv4 address:

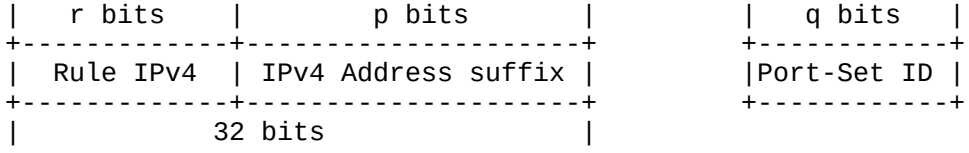


Figure 6: Shared IPv4 address

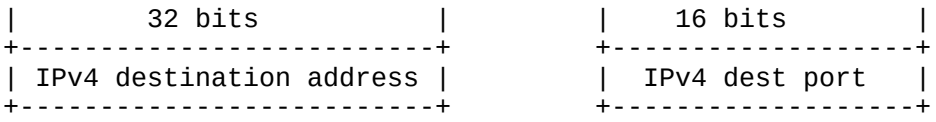
The length of r MAY be 32, with no part of the IPv4 address embedded in the EA bits. This results in a mapping with no dependence between the IPv4 address and the IPv6 address. In addition the length of o MAY be zero (no EA bits embedded in the End-User IPv6 prefix), meaning that also the PSID is provisioned using e.g., the DHCP option.

See Appendix A for an example of the Basic Mapping Rule.

5.3. Forwarding mapping rule (FMR)

The Forwarding Mapping Rule is optional, and used in mesh mode to enable direct CE to CE connectivity.

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



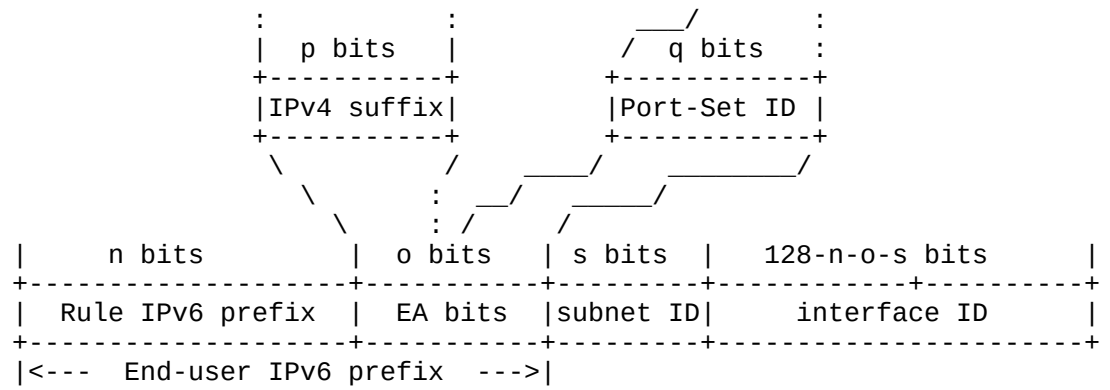


Figure 7: Derivation of MAP IPv6 address

See Appendix A for an example of the Forwarding Mapping Rule.

5.4. Destinations outside the MAP domain

IPv4 traffic between MAP nodes that are all within one MAP domain is encapsulated in IPv6, with the sender's MAP IPv6 address as the IPv6 source address and the receiving MAP node's MAP IPv6 address as the IPv6 destination address. To reach IPv4 destinations outside of the MAP domain, traffic is also encapsulated in IPv6, but the destination IPv6 address is set to the configured IPv6 address of the MAP BR.

On the CE, the path to the BR can be represented as a point to point IPv4 over IPv6 tunnel [RFC2473] with the source address of the tunnel being the CE's MAP IPv6 address and the BR IPv6 address as the remote tunnel address. When MAP is enabled, a typical CE router will install a default IPv4 route to the BR.

The BR forwards traffic received from the outside to CE's using the normal MAP forwarding rules.

6. The IPv6 Interface Identifier

The Interface identifier format of a MAP node is described below.

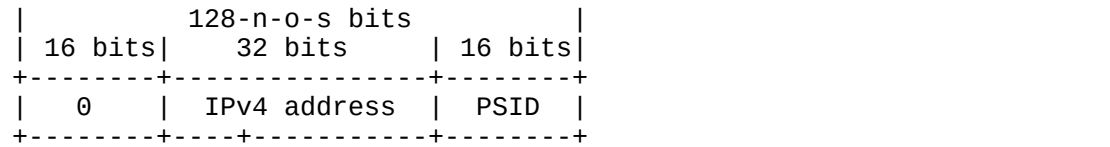


Figure 8

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

If the End-user IPv6 prefix length is larger than 64, the most significant parts of the interface identifier is overwritten by the prefix.

7. MAP Configuration

For a given MAP domain, the BR and CE MUST be configured with the following MAP elements. The configured values for these elements are identical for all CEs and BRs within a given MAP domain.

- The Basic Mapping Rule and optionally the Forwarding Mapping Rules, including the Rule IPv6 prefix, Rule IPv4 prefix, and Length of EA bits

- Hub and spoke mode or Mesh mode. (If all traffic should be sent to the BR, or if direct CE to CE traffic should be supported).

In addition the MAP CE MUST be configured with the IPv6 address(es) of the MAP BR (Section 5.4).

7.1. MAP CE

The MAP elements are set to values that are the same across all CEs within a MAP domain. The values may be configured in a variety of manners, including provisioning methods such as the Broadband Forum's "TR-69" Residential Gateway management interface, an XML-based object retrieved after IPv6 connectivity is established, or manual configuration by an administrator. IPv6 DHCP options for MAP configuration is defined in [I-D.ietf-softwire-map-dhcp]. Other configuration and management methods may use the format described by this option for consistency and convenience of implementation on CEs that support multiple configuration methods.

The only remaining provisioning information the CE requires in order to calculate the MAP IPv4 address and enable IPv4 connectivity is the IPv6 prefix for the CE. The End-user IPv6 prefix is configured as part of obtaining IPv6 Internet access.

The MAP provisioning parameters, and hence the IPv4 service itself, are tied to the associated End-user IPv6 prefix lifetime; thus, the MAP service is also tied to this in terms of authorization, accounting, etc.

A single MAP CE MAY be connected to more than one MAP domain, just as any router may have more than one IPv4-enabled service provider facing interface and more than one set of associated addresses assigned by DHCP. Each domain a given CE operates within would require its own set of MAP configuration elements and would generate its own IPv4 address. Each MAP domain requires a distinct End-user IPv6 prefix.

The MAP DHCP option is specified in [I-D.ietf-softwire-map-dhcp].

7.2. MAP BR

The MAP BR MUST be configured with corresponding mapping rules for each MAP domain which it is acting as BR for.

For increased reliability and load balancing, the BR IPv6 address MAY be an anycast address shared across a given MAP domain. As MAP is stateless, any BR may be used at any time. If the BR IPv6 address is anycast the relay MUST use this anycast IPv6 address as the source address in packets relayed to CEs.

Since MAP uses provider address space, no specific routes need to be advertised externally for MAP to operate, neither in IPv6 nor IPv4 BGP. However, if anycast is used for the MAP IPv6 relays, the anycast addresses must be advertised in the service provider’s IGP.

8. Forwarding Considerations

Figure 1 depicts the overall MAP architecture with IPv4 users (N and M) networks connected to a routed IPv6 network.

MAP uses Encapsulation mode as specified in [RFC2473].

For a shared IPv4 address, a MAP CE forwarding IPv4 packets from the LAN performs NAT44 functions first and creates appropriate NAT44 bindings. The resulting IPv4 packets MUST contain the source IPv4 address and source transport identifiers specified by the MAP provisioning parameters. The IPv4 packet is forwarded using the CE’s MAP forwarding function. The IPv6 source and destination addresses MUST then be derived as per Section 5 of this draft.

8.1. Receiving Rules

A MAP CE receiving an IPv6 packet to its MAP IPv6 address sends this packet to the CE's MAP function where it is decapsulated. The resulting IPv4 packet is then forwarded to the CE's NAT44 function where it is handled according to the NAT's translation table.

A MAP BR receiving IPv6 packets selects a best matching MAP domain rule (Rule IPv6 prefix) based on a longest address match of the packet's IPv6 source address, as well as a match of the packet destination address against the configured BR IPv6 address(es). The selected MAP rule allows the BR to determine the EA-bits from the source IPv6 address.

To prevent spoofing of IPv4 addresses, any MAP node (CE and BR) MUST perform the following validation upon reception of a packet. First, the embedded IPv4 address or prefix, as well as PSID (if any), are extracted from the source IPv6 address using the matching MAP rule. These represent the range of what is acceptable as source IPv4 address and port. Secondly, the node extracts the source IPv4 address and port from the IPv4 packet encapsulated inside the IPv6 packet. If they are found to be outside the acceptable range, the packet MUST be silently discard and a counter incremented to indicate that a potential spoofing attack may be underway. The source validation checks just described are not done for packets whose source IPv6 address is that of the BR (BR IPv6 address).

By default, the CE router MUST drop packets received on the MAP virtual interface (i.e., after decapsulation of IPv6) for IPv4 destinations not for its own IPv4 shared address, full IPv4 address or IPv4 prefix.

8.2. ICMP

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

If a MAP CE receives an ICMP message having ICMP identifier field in ICMP header, NAT44 in the MAP CE MUST rewrite this field to a specific value assigned from the port set. BR and other CEs must handle this field similar to the port number in the TCP/UDP header upon receiving the ICMP message with ICMP identifier field.

If a MAP node receives an ICMP error message without the ICMP identifier field for errors that is detected inside a IPv6 tunnel, a node should relay the ICMP error message to the original source. This behavior SHOULD be implemented conforming to the section 8 of [RFC2473].

8.3. Fragmentation and Path MTU Discovery

Due to the different sizes of the IPv4 and IPv6 header, handling the maximum packet size is relevant for the operation of any system connecting the two address families. There are three mechanisms to handle this issue: Path MTU discovery (PMTUD), fragmentation, and transport-layer negotiation such as the TCP Maximum Segment Size (MSS) option [RFC0897]. MAP uses all three mechanisms to deal with different cases.

8.3.1. Fragmentation in the MAP domain

Encapsulating an IPv4 packet to carry it across the MAP domain will increase its size (typically by 40 bytes). It is strongly recommended that the MTU in the MAP domain be well managed and that the IPv6 MTU on the CE WAN side interface be set so that no fragmentation occurs within the boundary of the MAP domain.

Fragmentation on MAP domain entry is described in section 7.2 of [RFC2473].

The use of an anycast source address could lead to an ICMP error message generated on the path being sent to a different BR. Therefore, using dynamic tunnel MTU Section 6.7 of [RFC2473] is subject to IPv6 Path MTU black-holes. A MAP BR using an anycast source address SHOULD NOT by default use Path MTU discovery across the MAP domain.

Multiple BRs using the same anycast source address could send fragmented packets to the same CE at the same time. If the fragmented packets from different BRs happen to use the same fragment ID, incorrect reassembly might occur. See [RFC4459] for an analysis of the problem. Section 3.4 suggests solving the problem by fragmenting the inner packet.

8.3.2. Receiving IPv4 Fragments on the MAP domain borders

Forwarding of an IPv4 packet received from the outside of the MAP domain requires the IPv4 destination address and the transport protocol destination port. The transport protocol information is only available in the first fragment received. As described in section 5.3.3 of [RFC6346] a MAP node receiving an IPv4 fragmented packet from outside has to reassemble the packet before sending the packet onto the MAP link. If the first packet received contains the transport protocol information, it is possible to optimize this behavior by using a cache and forwarding the fragments unchanged. Implementers of MAP should be aware that there are a number of well-known attacks against IP fragmentation; see [RFC1858] and [RFC3128].

Implementers should also be aware of additional issues with reassembling packets at high rates, as described in [RFC4963].

8.3.3. Sending IPv4 fragments to the outside

If two IPv4 host behind two different MAP CEs with the same IPv4 address sends fragments to an IPv4 destination host outside the domain, those hosts may use the same IPv4 fragmentation identifier, resulting in incorrect reassembly of the fragments at the destination host. Given that the IPv4 fragmentation identifier is a 16 bit field, it could be used similarly to port ranges. A MAP CE could rewrite the IPv4 fragmentation identifier to be within its allocated port-set, if the resulting fragment identifier space was large enough related to the rate fragments was sent. However, splitting the identifier space in this fashion would increase the probability of reassembly collision for all connections through the CPE. See also [RFC6864]

9. NAT44 Considerations

The NAT44 implemented in the MAP CE SHOULD conform with the behavior and best current practice documented in [RFC4787], [RFC5508], and [RFC5382]. In MAP address sharing mode (determined by the MAP domain /rule configuration parameters) the operation of the NAT44 MUST be restricted to the available port numbers derived via the basic mapping rule.

10. IANA Considerations

This specification does not require any IANA actions.

11. Security Considerations

Spoofing attacks: With consistency checks between IPv4 and IPv6 sources that are performed on IPv4/IPv6 packets received by MAP nodes, MAP does not introduce any new opportunity for spoofing attacks that would not already exist in IPv6.

Denial-of-service attacks: In MAP domains where IPv4 addresses are shared, the fact that IPv4 datagram reassembly may be necessary introduces an opportunity for DOS attacks. This is inherent to address sharing, and is common with other address sharing approaches such as DS-Lite and NAT64/DNS64. The best protection against such attacks is to accelerate IPv6 deployment, so that, where MAP is supported, it is less and less used.

Routing-loop attacks: This attack may exist in some automatic tunneling scenarios are documented in [RFC6324]. They cannot

exist with MAP because each BRs checks that the IPv6 source address of a received IPv6 packet is a CE address based on Forwarding Mapping Rule.

Attacks facilitated by restricted port set: From hosts that are not subject to ingress filtering of [RFC2827], some attacks are possible by an attacker injecting spoofed packets during ongoing transport connections ([RFC4953], [RFC5961], [RFC6056]. The attacks depend on guessing which ports are currently used by target hosts, and using an unrestricted port-set is preferable, i.e., using native IPv6 connections that are not subject to MAP port range restrictions. To minimize this type of attacks when using a restricted port-set, the MAP CE's NAT44 filtering behavior SHOULD be "Address-Dependent Filtering [RFC4787], Section 5. Furthermore, the MAP CEs SHOULD use a DNS transport proxy [RFC5625] function to handle DNS traffic, and source such traffic from IPv6 interfaces not assigned to MAP.

[RFC6269] outlines general issues with IPv4 address sharing.

12. Contributors

This document is the result of the IETF Softwire MAP design team effort and numerous previous individual contributions in this area:

Chongfeng Xie (China Telecom)
Room 708, No.118, Xizhimennei Street Beijing 100035
People's Republic of China
Phone: +86-10-58552116
Email: xiechf@ctbri.com.cn

Qiong Sun (China Telecom)
Room 708, No.118, Xizhimennei Street Beijing 100035
People's Republic of China
Phone: +86-10-58552936
Email: sunqiong@ctbri.com.cn

Gang Chen (China Mobile)
53A,Xibianmennei Ave. Beijing 100053
People's Republic of China
Email: chengang@chinamobile.com

Yu Zhai
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University

Beijing 100084
People's Republic of China
Email: jacky.zhai@gmail.com

Wentao Shang (CERNET Center/Tsinghua University)
Room 225, Main Building, Tsinghua University Beijing 100084
People's Republic of China
Email: wentaoshang@gmail.com

Guoliang Han (CERNET Center/Tsinghua University)
Room 225, Main Building, Tsinghua University Beijing 100084
People's Republic of China
Email: bupthgl@gmail.com

Rajiv Asati (Cisco Systems)
7025-6 Kit Creek Road Research Triangle Park NC 27709 USA
Email: rajiva@cisco.com

13. Acknowledgments

This document is based on the ideas of many, including Masakazu Asama, Mohamed Boucadair, Gang Chen, Maoke Chen, Wojciech Dec, Xiaohong Deng, Jouni Korhonen, Tomasz Mrugalski, Jacni Qin, Chunfa Sun, Qiong Sun, and Leaf Yeh. The authors want in particular to recognize Remi Despres, who has tirelessly worked on generalized mechanisms for stateless address mapping.

The authors would like to thank Lichun Bao, Guillaume Gottard, Dan Wing, Jan Zorz, Necj Scoberne, Tina Tsou, Kristian Poscic, and especially Tom Taylor and Simon Perreault for the thorough review and comments of this document. Useful IETF Last Call comments were received from Brian Weis and Lei Yan.

14. References

14.1. Normative References

[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.

[RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, August 2009.

14.2. Informative References

[I-D.ietf-softwire-map-deployment]
Qiong, Q., Chen, M., Chen, G., Tsou, T., and S. Perreault, "Mapping of Address and Port (MAP) - Deployment Considerations", draft-ietf-softwire-map-deployment-03 (work in progress), October 2013.

[I-D.ietf-softwire-map-dhcp]
Mrugalski, T., Troan, O., Dec, W., Bao, C., leaf.yeh.sdo@gmail.com, l., and X. Deng, "DHCPv6 Options for configuration of Softwire Address and Port Mapped Clients", draft-ietf-softwire-map-dhcp-06 (work in progress), November 2013.

[I-D.ietf-softwire-stateless-4v6-motivation]
Boucadair, M., Matsushima, S., Lee, Y., Bonness, O., Borges, I., and G. Chen, "Motivations for Carrier-side Stateless IPv4 over IPv6 Migration Solutions", draft-ietf-softwire-stateless-4v6-motivation-05 (work in progress), November 2012.

[RFC0897] Postel, J., "Domain name system implementation schedule", RFC 897, February 1984.

[RFC1858] Ziemba, G., Reed, D., and P. Traina, "Security Considerations for IP Fragment Filtering", RFC 1858, October 1995.

[RFC1933] Gilligan, R. and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 1933, April 1996.

[RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, March 1999.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

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

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

[RFC3128] Miller, I., "Protection Against a Variant of the Tiny Fragment Attack (RFC 1858)", RFC 3128, June 2001.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, April 2006.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

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

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

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

[RFC4963] Heffner, J., Mathis, M., and B. Chandler, "IPv4 Reassembly Errors at High Data Rates", RFC 4963, July 2007.

[RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.

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

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

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

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

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

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

[RFC6250] Thaler, D., "Evolution of the IP Model", RFC 6250, May 2011.

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

[RFC6324] Nakibly, G. and F. Templin, "Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", RFC 6324, August 2011.

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

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, August 2011.

[RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.

[RFC6864] Touch, J., "Updated Specification of the IPv4 ID Field", RFC 6864, February 2013.

Appendix A. Examples

Example 1 - Basic Mapping Rule

```
      Given the MAP domain information and an IPv6 address of
      an endpoint:

      End-user IPv6 prefix: 2001:db8:0012:3400::/56
      Basic Mapping Rule:   {2001:db8:0000::/40 (Rule IPv6 prefix),
                             192.0.2.0/24 (Rule IPv4 prefix),
                             16 (Rule EA-bits length)}
      PSID length:          (16 - (32 - 24) = 8. (Sharing ratio of 256)
      PSID offset:          6 (default)

      A MAP node (CE or BR) can via the BMR, or equivalent FMR,
      determine the IPv4 address and port-set as shown below:

      EA bits offset:       40
      IPv4 suffix bits (p)  Length of IPv4 address (32) -
                            IPv4 prefix length (24) = 8
      IPv4 address:         192.0.2.18 (0xc0000212)
      PSID start:           40 + p = 40 + 8 = 48
      PSID length:          o - p = (56 - 40) - 8 = 8
      PSID:                 0x34

      Available ports (63 ranges) : 1232-1235, 2256-2259, ..... ,
                                    63696-63699, 64720-64723

      The BMR information allows a MAP CE to determine (complete)
      its IPv6 address within the indicated IPv6 prefix.

      IPv6 address of MAP CE:  2001:db8:0012:3400:0000:c000:0212:0034
```

Example 2 - BR:

Another example can be made of a MAP BR,
configured with the following FMR when receiving a packet
with the following characteristics:

```
IPv4 source address:       1.2.3.4 (0x01020304)
IPv4 source port:          80
IPv4 destination address:  192.0.2.18 (0xc0000212)
IPv4 destination port:     1232

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

IPv6 address of MAP BR:              2001:db8:ffff::1
```

The above information allows the BR to derive as follows
the mapped destination IPv6 address for the corresponding
MAP CE, and also the mapped source IPv6 address for
the IPv4 source address.

```
IPv4 suffix bits (p):  32 - 24 = 8 (18 (0x12))
PSID length:           8
PSID:                  0x34 (1232)
```

The resulting IPv6 packet will have the following key fields:

```
IPv6 source address:       2001:db8:ffff::1
IPv6 destination address:  2001:db8:0012:3400:0000:c000:0212:0034
```

Example 3 - Forwarding Mapping Rule:

An IPv4 host behind the MAP CE (addressed as per the previous
examples) corresponding with IPv4 host 1.2.3.4 will have its
packets encapsulated by IPv6 using the IPv6 address of the BR
configured on the MAP CE as follows:

```
IPv6 address of BR:         2001:db8:ffff::1
IPv4 source address:        192.0.2.18
IPv4 destination address:   1.2.3.4
IPv4 source port:           1232
IPv4 destination port:      80
MAP CE IPv6 source address: 2001:db8:0012:3400:0000:c000:0212:0034
IPv6 destination address:   2001:db8:ffff::1
```

Example 4 - Rule with no embedded address bits and no address sharing

```
End-User IPv6 prefix: 2001:db8:0012:3400::/56
Basic Mapping Rule:   {2001:db8:0012:3400::/56 (Rule IPv6 prefix),
                       192.0.2.18/32 (Rule IPv4 prefix),
                       0 (Rule EA-bits length)}
PSID length:          0 (Sharing ratio is 1)
PSID offset:          n/a
```

A MAP node (CE or BR) can via the BMR or equivalent FMR, determine the IPv4 address and port-set as shown below:

```
EA bits offset:       0
IPv4 suffix bits (p): Length of IPv4 address (32) -
                      IPv4 prefix length (32) = 0
IPv4 address:         192.0.2.18 (0xc0000212)
PSID start:           0
PSID length:          0
PSID:                 null
```

The BMR information allows a MAP CE also to determine (complete) its full IPv6 address by combining the IPv6 prefix with the MAP interface identifier (that embeds the IPv4 address).

IPv6 address of MAP CE: 2001:db8:0012:3400:0000:c000:0212:0000

Example 5 - Rule with no embedded address bits and address sharing (sharing ratio 256)

```
End-User IPv6 prefix: 2001:db8:0012:3400::/56
Basic Mapping Rule:   {2001:db8:0012:3400::/56 (Rule IPv6 prefix),
                       192.0.2.18/32 (Rule IPv4 prefix),
                       0 (Rule EA-bits length)}
PSID length:          8. (From DHCP. Sharing ratio of 256)
PSID offset:          6  (Default)
PSID       :          0x34 (From DHCP.)
```

A MAP node can via the Basic Mapping Rule determine the IPv4 address and port-set as shown below:

```
EA bits offset:        0
IPv4 suffix bits (p):  Length of IPv4 address (32) -
                       IPv4 prefix length (32) = 0
IPv4 address:          192.0.2.18 (0xc0000212)
PSID offset:           6
PSID length:           8
PSID:                  0x34

Available ports (63 ranges) : 1232-1235, 2256-2259, ..... ,
                              63696-63699, 64720-64723
```

The Basic Mapping Rule information allows a MAP CE also to determine (complete) its full IPv6 address by combining the IPv6 prefix with the MAP interface identifier (that embeds the IPv4 address and PSID).

IPv6 address of MAP CE: 2001:db8:0012:3400:0000:c000:0212:0034

Note that the IPv4 address and PSID is not derived from the IPv6 prefix assigned to the CE, but provisioned separately using e.g., DHCP.

Appendix B. A More Detailed Description of the Derivation of the Port Mapping Algorithm

This Appendix describes how the port mapping algorithm described in Section 5.1 was derived. The algorithm is used in domains whose rules allow IPv4 address sharing.

The basic requirement for a port mapping algorithm is that the port-sets it assigns to different MAP CEs MUST be non-overlapping. A number of other requirements guided the choice of the algorithm:

- In keeping with the general MAP algorithm the port-set MUST be derivable from a port-set identifier (PSID) that can be embedded in the End-user IPv6 prefix.

- The mapping MUST be reversible, such that, given the port number, the PSID of the port-set to which it belongs can be quickly derived.

- The algorithm MUST allow a broad range of address sharing ratios.

- It SHOULD be possible to exclude subsets of the complete port numbering space from assignment. Most operators would exclude the system ports (0-1023). A conservative operator might exclude all but the transient ports (49152-65535).

- The effect of port exclusion on the possible values of the End-user IPv6 prefix (i.e., due to restrictions on the PSID value) SHOULD be minimized.

- For administrative simplicity, the algorithm SHOULD allocate the the same or almost the same number of ports to each CE sharing a given IPv4 address.

The two extreme cases that an algorithm satisfying those conditions might support are: (1) the port numbers are not contiguous for each PSID, but uniformly distributed across the allowed port range; (2) the port numbers are contiguous in a single range for each PSID. The port mapping algorithm proposed here is called the Generalized Modulus Algorithm (GMA) and supports both these cases.

For a given IPv4 address sharing ratio (R) and the maximum number of contiguous ports (M) in a port-set, the GMA is defined as:

a. The port numbers (P) corresponding to a given PSID are generated by:

(1) ... P = (R * M) * i + M * PSID + j

where i and j are indices and the ranges of i, j, and the PSID are discussed in a moment.

b. For any given port number P, the PSID is calculated as:

(2) ... PSID = trunc((P modulo (R * M)) / M)

where trunc() is the operation of rounding down to the nearest integer.

Formula (1) can be interpreted as follows. First, the available port space is divided into blocks of size R * M. Each block is divided into R individual ranges of length M. The index i in formula (1) selects a block, PSID selects a range within that block, and the index j selects a specific port value within the range. On the basis of this interpretation:

- i ranges from ceil(N / (R * M)) to trunc(65536/(R * M)) - 1, where ceil is the operation of rounding up to the nearest integer and N is the number of ports (e.g., 1024) excluded from the lower end of the range. That is, any block containing excluded values is discarded at the lower end, and if the final block has fewer than R * M values it is discarded. This ensures that the same number of ports is assigned to every PSID.

- PSID ranges from 0 to R - 1;

- j ranges from 0 to M - 1.

B.1. Bit Representation of the Algorithm

If R and M are powers of 2 (R = 2^k, M = 2^m), formula (1) translates to a computationally convenient structure for any port number represented as a 16-bit binary number. This structure is shown in Figure 9.

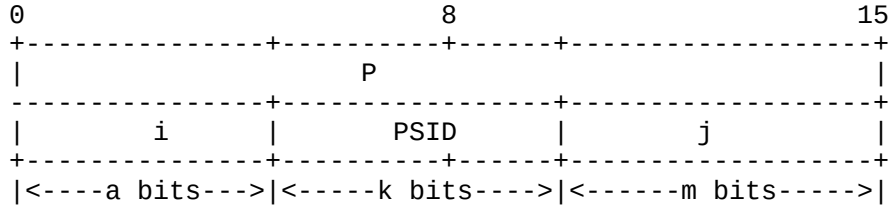


Figure 9: Bit Representation of a Port Number

As shown in the figure, the index value i of formula (1) is given by the first a = 16 - k - m bits of the port number. The PSID value is given by the next k bits, and the index value j is given by the last m bits.

Because the PSID is always in the same position in the port number and always the same length, different PSID values are guaranteed to generate different sets of port numbers. In the reverse direction,

the generating PSID can be extracted from any port number by a bit mask operation.

Note that when M and R are powers of 2, 65536 divides evenly by R * M. Hence the final block is complete and the upper bound on i is exactly 65536/(R * M) - 1. The lower bound on i is still the minimum required to ensure that the required set of ports is excluded. No port numbers are wasted through discarding of blocks at the lower end if block size R * M is a factor of N, the number of ports to be excluded.

As a final note, the number of blocks into which the range 0-65535 is being divided in the above representation is given by 2^a. Hence the case where a = 0 can be interpreted as one where the complete range has been divided into a single block, and individual port-sets are contained in contiguous ranges in that block. We cannot throw away the whole block in that case, so port exclusion has to be achieved by putting a lower bound equal to ceil(N / M) on the allowed set of PSID values instead.

B.2. GMA examples

For example, for R = 256, PSID = 0, offset: a = 6 and PSID length: k = 8 bits

Available ports (63 ranges) : 1024-1027, 2048-2051, ... ,
63488-63491, 64512-64515

Example 1: with offset = 6 (a = 6)

For example, for R = 64, PSID = 0, a = 0 (PSID offset = 0 and PSID length = 6 bits), no port exclusion:

Available ports (1 range) : 0-1023

Example 2: with offset = 0 (a = 0) and N = 0

Authors' Addresses

Ole Troan (editor)
Cisco Systems
Philip Pedersens vei 1
Lysaker 1366
Norway

Email: ot@cisco.com

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

Email: wdec@cisco.com

Xing Li
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
People's Republic of China

Email: xing@cernet.edu.cn

Congxiao Bao
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
People's Republic of China

Email: congxiao@cernet.edu.cn

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

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

Tetsuya Murakami
IP Infusion
1188 East Arques Avenue
Sunnyvale
USA

Email: tetsuya@ipinfusion.com

Tom Taylor (editor)
Huawei Technologies
Ottawa
Canada

Email: tom.taylor.stds@gmail.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 11, 2015

Q. Sun
China Telecom
M. Chen
BBIX
G. Chen
China Mobile
T. Tsou
Huawei Technologies
S. Perreault
Jive Communications
June 9, 2015

Mapping of Address and Port (MAP) - Deployment Considerations
draft-ietf-softwire-map-deployment-06

Abstract

This document describes when and how an operator uses the technique of Mapping of Address and Port (MAP) for the IPv4 residual deployment in the IPv6-dominant domain.

Status of this Memo

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

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

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

This Internet-Draft will expire on December 11, 2015.

Copyright Notice

Copyright (c) 2015 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. Case Studies 5
4. Deployment Consideration 7
4.1. Building the MAP Domain 7
4.1.1. MAP Deployment Model Planning 7
4.1.2. MAP Domain Planning 8
4.1.3. MAP Rule Provisioning 8
4.1.4. MAP DHCPv6 server deployment consideration 9
4.1.5. PSID Consideration 10
4.1.6. Addressing and Routing 10
4.1.7. MAP vs. MAP-T vs. 4rd 11
4.2. BR Settings 12
4.3. CE Settings 15
4.4. Supporting System 15
5. MAP Address Planning 17
5.1. Planning for Residual Deployment, a Step-by-step Guide . . 17
5.2. Remarks on Deployment Paradigms 19
6. Migration Methodology 21
6.1. Roadmap for MAP-based Solution 21
6.1.1. Start from Scratch 21
6.1.2. Coexiting Phases 21
6.1.3. Exit Strategy 21
6.2. Migration Mode 22
6.2.1. Passive Transition 22
6.2.2. Active Transition 22
7. IANA Considerations 23
8. Security Considerations 24
9. Contributors 25
10. Acknowledgements 26
11. References 27
11.1. Normative References 27
11.2. Informative References 27
Authors' Addresses 29

1. Introduction

IPv4 address exhaustion has become world-wide reality and the primary solution in the industry is to deploy IPv6-only networking. Meanwhile, having access to legacy IPv4 contents and services is a long-term requirement, will be so until the completion of the IPv6 transition. It demands sharing residual IPv4 address pools for IPv4 communications across the IPv6-only domain(s).

Mapping of Address and Port (MAP) [I-D.ietf-softwire-map] is designed in response to the requirement of stateless residual deployment. The term "residual deployment" refers to utilizing IPv4 addresses for IPv4 communications going across the IPv6 domain backbone. MAP assumes the IPv6-only backbone as the prerequisite of deployment so that native IPv6 services and applications are fully supported and encouraged. The statelessness of MAP ensures only moderate overhead is added to part of the network devices.

Residual deployment with MAP is new to most operators. This document is motivated to provide basic understanding on the usage of MAP, i.e., when and how an operator can do with MAP to meet its own operational requirements of IPv6 transition and its facility conditions, in the phase of IPv4 residual deployment. Potential readers of this document are those who want to know:

1. What are the requirements of MAP deployment ?

2. What technical options needs to be considered when deploying MAP, and how?

3. How does MAP impact on the address planning for both IPv6 and IPv4 pools?

4. How does MAP impact on daily network operations and administrations?

5. How do we migrate to IPv6-only network with the help of MAP?

Terminology of this document, unless it is intentionally specified, follows the definitions and abbreviations of [I-D.ietf-softwire-map].

Unless it is specifically specified, the deployment considerations and guidance proposed in this document are also applied to MAP-T [I-D.ietf-softwire-map-t], the translation variation of MAP, and 4rd [I-D.ietf-softwire-4rd], the reversible translation approach that aims to improve end-to-end consistency of double translation.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Case Studies

MAP can be deployed for large-scale carrier networks. There are typically two network models for broadband access service: one is to use PPPoE/PPPoA authentication method while the other is to use IPoE. The first one is usually applied to Residential network and SOHO networks. Subscribers in CPNs can access broadband network by PPP dial-up authentication. BRAS is the key network element which takes full responsibility of IP address assignment, user authentication, traffic aggregation, PPP session termination, etc. Then IP traffic is forwarded to Core Routers through Metro Area Network, and finally transited to Internet via Backbone network. The second network scenario is usually applied to large enterprise networks. Subscribers in CPNs can access broadband network by IPoE authentication. IP address is normally assigned by DHCP server, or static configuration.

In either case, a Customer Edge Router(CER) could obtain a prefix via prefix delegation procedure, and the hosts behind CER would get its own IPv6 addresses within the prefix through SLAAC or DHCPv6 statefully. A MAP CE would also obtain a set of MAP rules from DHCPv6 server.

Figure 1 depicts a generic model of stateless IPv4-over-IPv6 communication for broadband access services.

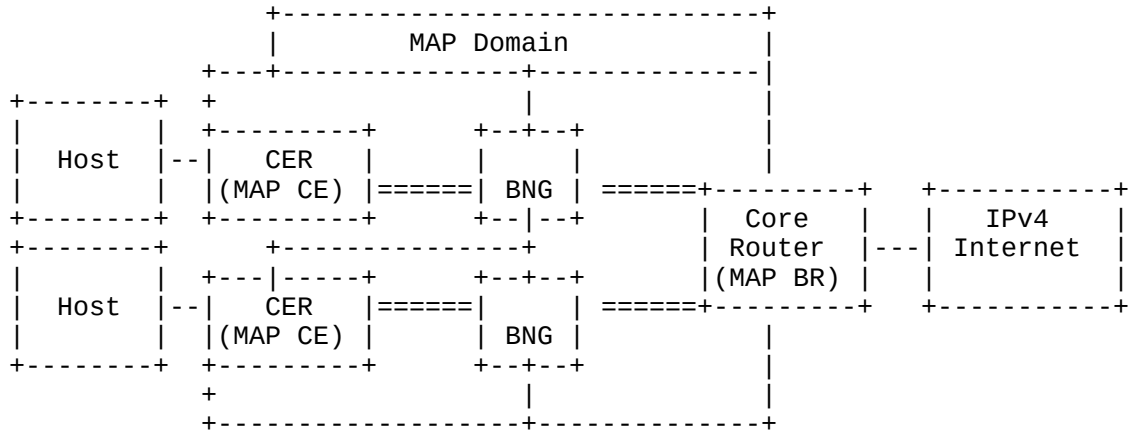


Figure 1: Stateless IPv4-over-IPv6 broadband access network architecture

When deploying MAP in home network, there can be two architecture: A. single ISP B. multihoming with two or more ISPs, sharing one CE. In the single ISP model, CE needs to communicate with only one MAP BR,

while in multihoming model CE has to communicate with multiple MAP BRs. Figure 2 [RFC7368] illustrates a typical case, where the home network has multiple connections to multiple providers or multiple logical connections to the same provider. In the multihoming model, a CE will be provisioned with multiple BMRs. Routing information will also be configured for multihoming; but detail of the routing configuration is out of the scope of this memo.

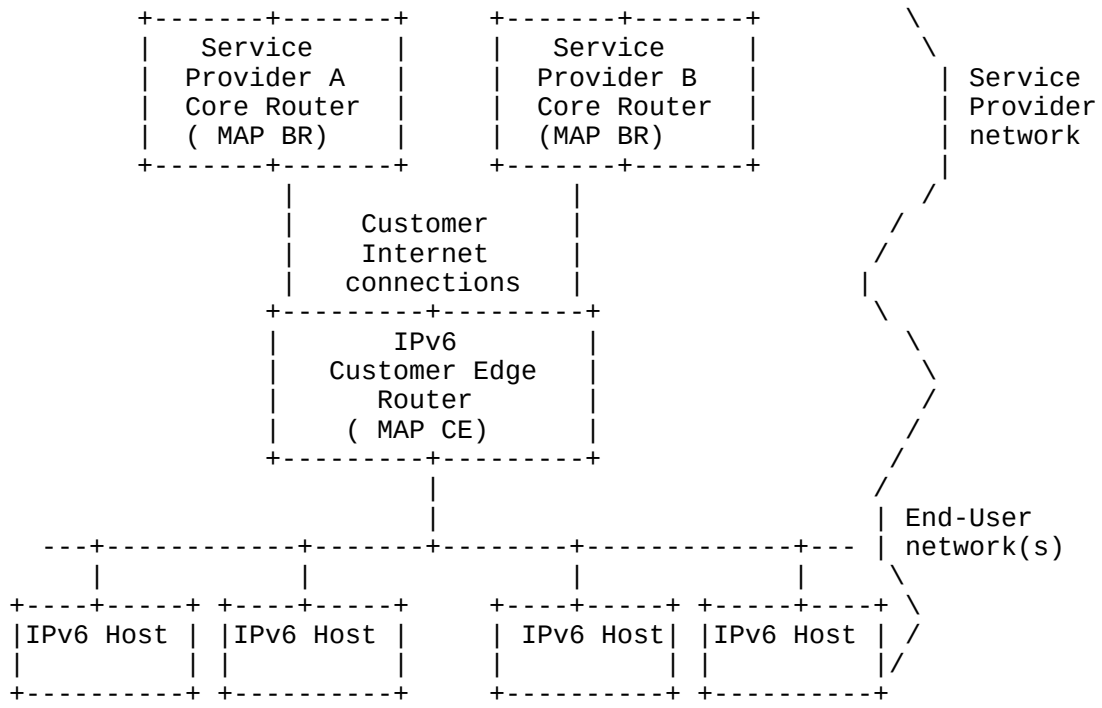


Figure 2: MAP multihoming

4. Deployment Consideration

4.1. Building the MAP Domain

When deploying stateless MAP in an operational network, a provider should firstly do MAP domain planning based on that existing network. According to the definition of [I-D.ietf-softwire-map], a MAP domain is a set of MAP CEs and BRs connected to the same virtual link. All CEs in the MAP domain are provisioned with a same set of MAP rules by MAP DHCPv6 server [I-D.ietf-softwire-map-dhcp]. There might be multiple BMRs in one MAP domain, e.g. in case of multi-ISP. A CE may be provisioned with multiple IPv6 prefix, which can be used to find the corresponding BMR via longest prefix match. As defined in [I-D.ietf-softwire-map-dhcp], a BMR should be provisioned together with a BR IPv6 address; the CE should maintain this binding, so that the mapping between BMR and BR is achieved which is useful in multi-ISP scenario. In in mesh mode, a longest-matching prefix lookup is done in the IPv4 routing table and the correct FMR is chosen.

Basically, operator should firstly determine its own deployment topology for MAP domain as described in Section 4.1.1, as different considerations apply for different deployment models. Next, MAP domain planning, MAP rule provision, addressing and routing, etc., for a MAP domain should be taken into consideration, as discussed in the sections following Section 4.1.1.

For the scenario where one CE is corresponding with multiple MAP border relays, it is possible that those MAP BRs belong to different MAP domains. The CE must pick up its own MAP rules and domain parameters in each domain. This is a typical case of multihoming. The MAP rules must have the information about BR(s) and information about the service types and the ISP.

4.1.1. MAP Deployment Model Planning

In order to do MAP domain planning, an operator should firstly make the decision to choose mesh or hub and spoke topology according to the operator’s network policy. In the hub and spoke topology, all traffic within the same MAP domain has to go through the BR, result in less optimal traffic flow; however, it simplifies CE processing since there is no need to do FMR lookup for each incoming packet. Moreover, it provides enhanced manageability as the BR can take full control of all the traffic. As a result, it is reasonable to deploy hub and spoke topology for a network with a relatively flat architecture.

In mesh topology, CE to CE traffic flows are optimized since they pass directly between the two nodes. Mesh topology is recommended

when CE to CE traffic is high and there are not too many MAP rules, say fewer than 10 MAP rules, in the given domain.

4.1.2. MAP Domain Planning

Stateless MAP offers advantages in terms of scalability, high reliability, etc. As a result, it is reasonable to plan for a larger MAP domain to accommodate more subscribers with fewer BRs. Moreover, a larger MAP domain will also be easier for management and maintenance. However, a larger MAP domain may also result in less optimized traffic in the hub and spoke case, where all traffic has to go through a remote BR. In addition, it may result in an increased number of MAP rules and highly centralized address management. Choosing appropriate domain coverage requires the evaluation of tradeoffs.

When multiple IPv4 subnets are deployed in one MAP domain, it is recommended to further divide the MAP domain into mutiple sub-domains, each with only one IPv4 subnet. This can simplify the MAP domain planning. But there can be a side effect that it will increase the traffic between BRs. Different subdomains could be distinguished by different Rule IPv4 prefixes. As stated previously, all CEs within the same MAP subdomain will have the same Rule IPv4 prefix, Rule IPv6 prefix and PSID parameters.

4.1.3. MAP Rule Provisioning

In stateless MAP, Mesh or Hub and Spoke communications can be achieved among CEs in one MAP domain in terms of assigning appropriate FMR(s) to CEs. We recommend ISP deploy the full Hub and Spoke topology or full mesh topology describe below to simplify the configuration of the DHCPv6 server.

4.1.3.1. Full Hub and Spoke Communication among CEs

In order to achieve the full communication in the Hub and Spoke topology, no FMR is assigned to CEs. In this topology, when a CE sends packets to another CE in the same MAP domain via BR, or using the DMR as FMR, the packets must go though BR before arriving at the destination. DMR is specific for MAP-T only.

4.1.3.2. Full Mesh Communication among CEs

By assigning all BMRs in MAP domain to each CE as FMRs, Mesh communications can be achieved among all CEs. In this case, when CE receives an IPv4 packet, it looks up for an appropriate FMR with a specific Rule IPv4 prefix which has the longest match with the IPv4 destination address.

4.1.3.3. Mesh or Hub/Spoke communication among some CEs

Mesh communications among some CEs along with Hub/Spoke communications among some other CEs can be achieved by which differentiated FMRs are assigned to CEs. For instance, as shown in Figure 3, since both CE1 and CE2 has rule 1 and rule2, the communication between CE1 and CE2 can go directly without going though associated BR (Mesh topology). However, for CE1 and CE3, since there are no rule for each other, the communication between CE1 and CE3 must go though BR before reaching peer each other (Hub/Spoke topology).

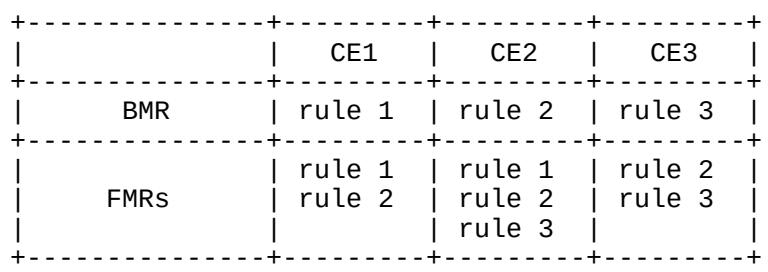


Figure 3:

4.1.4. MAP DHCPv6 server deployment consideration

All the CEs within a MAP domain will get a set of MAP rules by DHCPv6 server. Each Mapping Rule keeps a record of Rule IPv6 prefix, Rule IPv4 prefix and Rule EA-bits length. Section 5 would give a step by step example of how to calculate these parameters.

As the MAP is stateless, the deployment of DHCPv6 server is independent of MAP domain planning. So there are three possible cases:

MAP domain : DHCPv6 server = 1:1 This is the ideal solution that each MAP domain would have its own MAP DHCPv6 server. In this case, MAP DHCPv6 server only needs to configure parameters for the specific MAP domain. In this model, it is easy to achieve the configuration in MAP and no extra configuration requirement is needed.

MAP domain : DHCPv6 server = 1:N This might happen when DHCPv6 servers are deployed in a large MAP domain in a distributed manner. In this case, all these DHCPv6 servers should be configured with the same set of MAP rules for the MAP domain, including multiple BMRs, FMRs and DMRs.

MAP domain : DHCPv6 server = N:1 This might happen when MAP domain is relatively small and a single MAP DHCPv6 server is deployed in the network. In this case, multiple MAP domains should be distinguished based on CE's IPv6 prefix in different MAP domains.

4.1.5. PSID Consideration

If a provider would like to introduce differentiated address sharing ratios for different CEs, it is better to define multiple MAP sub-domains with different Rule IPv4 prefixes. In this way, MAP domain division is only a logical method, rather than a geographical one.

The default PSID offset(a) is chosen as 6 in [I-D.ietf-softwire-map] and this excludes the system ports (0-1023). For MAP, the initial part of the port number (the a-bits) cannot be zero (see Appendix B of [I-D.ietf-softwire-map].) As is shown in the section 3.2.4 of [I-D.tsou-softwire-port-set-algorithms-analysis], it is possible that a lower value of 'a' will give a higher sharing ratio and more than 1024 ports are excluded as a result, e.g. 'a' = 4 will exclude ports 0 - 4095. The value of 'a' should be made explicitly configurable by operators.

With regard to PSID format, both continuous and non-continuous port set can be supported in GMA algorithm. Non-continuous port set has the advantage of better UPnP friendly, while continuous port set is the simplest way to implement. Since PSID format should be supported not only in CPEs, BRs and DHCPv6 server, but also in other sustaining systems as well, e.g. traffic logging system, user management system, a provider should make the decision based on a comprehensive investigation on its demand and the capabilities of existing equipments.

Note that some ISPs may need to offer services in a MAP domain with a shared address, e.g. there are hosts FTP server under CEs. The service provisioning may require well-know port range (i.e. port range belong to 0-1023). MAP would provide operators with an option to generate a port range including those in 0-1023. Afterwards, operators could decide to assign it to any requesting user. However, if the port-set is too small, it is not suggested to assign one with only the port set 0˜1023 or even less. Considerable non-well-known ports are surely needed. Another easier approach is assigning a dedicated IPv4 address to such a CE if the demand really exists.

4.1.6. Addressing and Routing

In MAP addressing, it should follow the MAP rule planning in the MAP domain.

For IPv4 addressing, since the number of scattered IPv4 address prefixes would be equal to the number of FMR rules within a MAP domain, one should choose as large IPv4 address pool as possible to reduce the number of FMR rules. For IPv6 address, the Rule IPv6 prefixes should be equal to the end user IPv6 prefix in MAP domain.

If ISP has a /24 rule IPv4 prefix with sharing ratio of 64 gives 16000 customers, and a /16 rule IPv4 prefix supports 4 million customer. If up the sharing ratio to 256, 64000 and 16 million customers can be supports respectively. For the ISP who has scattered IPv4 address prefixes, in order to reduce the number of FMRs, according to needs of ports they can divide different classes. For instance, for the enterprise customers class which need many ports to use, provision them the BMR with low sharing ratio while for the private customers class which don’t need so many ports provision them the BMR with high sharing ratio.

For MAP routing, there are no IPv4 routes exported to IPv6 networks.

4.1.7. MAP vs. MAP-T vs. 4rd

Basically, encapsulation provides an architectural building block of virtual link where the underlay behavior is fully hidden, while translation does a delivery participating into the end-to-end transferring path where behaviors are exposed. It is reflected in the following aspects.

1. Option header

If translation or 4rd ’reversible translation’ is applied, IPv4 options at the IP layer are not translated according to [RFC791][RFC2460], and packets with those options MUST be dropped by Domain-entry nodes, and return ICMPv4 error messages to signal IPv4-option incompatibility. This limitation is acceptable because there are a lot firewalls in current IPv4 Internet also filter IPv4 packets

2. ICMP

Some IPv4 ICMP codes do not have a corresponding codes in ICMPv6, a detailed analysis on the double translation behavior suggest that some ICMPv4 messages, when they are translated to ICMPv6 and back to ICMPv4 across the IPv6 domain, the accuracy might be sacrificed to some extent. Encapsulation keeps the full transparency of ICMPv4 messages.

Reversible translation approach of 4rd, however, does not translate ICMPv4 messages into ICMPv6 version. Instead, it treats ICMP as same as a transport layer protocol data unit. This behavior is similar to

the encapsulation and keeps ICMP end-to-end transparency as well.

In either the encapsulation or translation mode, if an intermediate node generates an ICMPv6 error message, it should be converted into ICMPv4 version and returned to the source with a special source address and following the behavior specified in [RFC6791]. However, the behavior and semantics of the translation from ICMPv6 to ICMPv4 is different among encapsulation, translation and 4rd reversible translation approaches. Encapsulation treats routing error in the IPv6 domain as an (virtual)link error between the tunnel end points, while translation translate IPv6 routing error into corresponding IPv4 version, and 4rd, however, behaves according to whether the Tunnel Traffic Class option is set. The TTL behavior also reflect the differences among different approaches, which is worth paying attention to for the operating engineers. MAP-T translator is compatible with single translation approach.

3. PMTU and fragmentation

Both translation mode and encapsulation mode have PMTU and fragmentation problem. [RFC6145] discusses the problem in details for the translation, while [RFC2473] could be a reference on the issue in encapsulation.

4.2. BR Settings

1. BR placement

BR placement has important impacts on the operation of a MAP domain.

A first concern should be the avoidance of "triangle routing". In hub and spoke mode, all traffic will be routed through BR which may increase the path from the CE to an IPv4 peer. This can be accomplished easily by placing the BR close to the CE, such that the length of the path from the CE to the BR is minimized.

However, minimizing the CE-BR path would ignore a second concern, that of minimizing IPv4 operations. An ISP deploying MAP will probably want to focus on IPv6 operations, while keeping IPv4 operational expenditures to a minimum. This would imply that the size of the IPv4 network that the ISP has to administer would be kept to a minimum. Placing the BR near the CE means that the length of the IPv4 network between the BR and the IPv4 Internet would be longer.

Moreover, in case where the set of CEs is geographically dispersed, multiple BRs would be needed, which would further enlarge the IPv4 network that the ISP has to maintain.

Therefore, we offer the following guideline: BRs should be placed as close to the border with the IPv4 Internet as possible while keeping triangle routing to a minimum. Regional POPs should probably be considered as potential candidates.

Note also that MAP being stateless, asymmetric routing to/from the IPv4 Internet is natively supported and therefore no path-pinning mechanisms have to be additionally implemented.

Anycast can be used to let the network pick BR closest to a CE for traffic exiting the MAP domain. This is accomplished by provisioning a Default Mapping Rule containing an anycast IPv6 address or prefix. Operationally, this allows incremental deployment of BRs in strategic locations without modifying the provisioning system's configuration. CE's close to a newly-deployed BR will automatically start using it. The BR MUST participate in a dynamic IGP so that this can work automatically.

2. Reliability Considerations

Reliability of MAP is derived in major part from its statelessness. This means that MAP can benefit from the usual methods of Internet reliability.

Anycast, already mentioned in section 4.2.1, can be used to ensure reliability of traffic from CE to BR. Since there can be only one Default Mapping Rule per MAP domain, traffic from CE to BR will always use the same destination address. When this address is anycast, reliability is greatly increased. If a BR goes down, it stops advertising the IPv6 anycast address, and traffic is automatically re-routed to other BRs; the BR should also withdraw the routes for traffic from BR to CE, or the upstream routers connected to the BR should dynamically change the routes when it detects the failure of a BR, otherwise there will be a routing blackhole. For this mechanism to work correctly, it is crucial that the anycast route announcement be very closely tied to BR availability. See [RFC4786] for best current practices on the operation of anycast services. In practice, Equal-cost multi-path (ECMP) can be used to achieve active/active configuration. Operator can also increase the metric for one BR to have active/standby.

For reliability within a single link can be achieved with the help of a redundancy protocol such as VRRP [RFC5798]. This allows operation of a pair of BRs in active/standby configuration. No state needs to be shared for the operation of MAP, so there is no need to keep the standby node in a "warm" state: as long as it is up and ready to take over the virtual IPv6 address, quick failover can be achieved. This makes the pair behave as a single, much more reliable node, with less

reliance on quick routing protocol convergence for reliability.

It is expected that production-quality MAP deployments will make use of both anycast and a redundancy protocol such as VRRP.

3. MTU/Fragmentation

If the MTU is well-managed such that the IPv6 MTU on the CE WAN side interface is set so that no fragmentation occurs within the boundary of the MAP domain, then the Tunnel MTU can be set to the known IPv6 MTU minus the size of the encapsulating IPv6 header (40 bytes). For example, if the IPv6 MTU is known to be 1500 bytes, the Tunnel MTU might be set to 1460 bytes. Without more specific information, the Tunnel MTU SHOULD default to 1240 bytes.

BRs using an anycast address as source can cause problems. If traffic sent by a BR with a source anycast address causes an ICMP error to be returned, that error packet's destination address will be an anycast address, meaning that a different BR might receive it. In the case of a Too Big ICMP error, this could cause a path MTU discovery black hole. Another possible problem could occur if fragmented packets from different BRs using the same anycast address as source happen to contain the same fragment ID. This would break fragment reassembly. Since there is still no simple way to solve it completely, it is recommended to increase the MTU of the IPv6 network so that no fragmentation and Too Big ICMP error occurs.

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

On the other hand, in MAP domains where IPv4 addresses are shared, BRs and CEs may have to encapsulate or translate IPv4 packets whose IPv6 destinations depend on destination ports. Precautions are needed, due to the fact that the destination port of a fragmented datagram is available only in its first fragment. A sufficient precaution consists in reassembling each datagram received in multiple packets, and to treat it as though it would have been received in single packet. This function is such that MAP is in this case stateful at the IP layer. (This is common with DS-lite and NAT64/DNS64 which, in addition, are stateful at the transport layer.) At domain entrance, this ensures that all pieces of all received IPv4 datagrams go to the right IPv6 destinations.

4.3. CE Settings

1. bridging vs. routing

In routing manner, the CE runs a standard NAT44 [RFC3022] using the allocated public address as external IP and ports via DHCPv6 option. When receiving an IPv4 packet with private source address from its end hosts, it performs NAT44 function by translating the source address into public and selecting a port from the allocated port-set. Then it encapsulates/translates (depending on whether MAP-E or MAP-T is in use) the packet with the concentrator's IPv6 address as destination IPv6 address, and forwards it to the concentrator. When receiving an IPv6 packet from the concentrator, the initiator decapsulates/translates the IPv6 packet to get the IPv4 packet with public destination IPv4 address. Then it performs NAPT44 function and translates the destination address into private one based on the entry in NAT state table in the CE.

The CE is responsible for performing ALG functions (e.g., SIP, FTP), as well as supporting NAT Traversal mechanisms (e.g., UPnP, NAT-PMP, manual mapping configuration). This is no different from the standard IPv4 NAT today.

For the bridging manner, end host would run a software performing CE functionalities. In this case, end host gets public address directly. It is also suggested that the host run a local NAT to map randomly generated ports into the restricted, valid port-set. Another solution is to have the IP stack to only assign ports within the restricted, valid range to applications. Either way the host guarantees that every source port number in the outgoing packets falls into the allocated port-set.

2. CE-initiated application

CE-initiated case is applied for situations where applications run on CE directly. If the application in CE use the public address directly, it might conflict with other CEs. So it is highly suggested that CE should also run a local NAT to map a private address to public address in CE. In this way, the CE IPv4 address passed to local applications would be conflict with other CEs.

4.4. Supporting System

1. Lawful Intercept

Sharing IPv4 addresses among multiple CEs is susceptible to issues related to lawful intercept. For details, see [RFC6269] section 12.

2. Traffic Logging

It is always possible for a service provider that operates a MAP domain to determine the IPv6 prefix associated with a MAP IPv4 address (and port number in case of a shared address). This mapping is static, and it is therefore unnecessary to log every IPv4 address assignment. However, changes in that static mapping, such as rule changes in the provisioning system, need to be logged in order to be able to know the mapping at any point in time.

Sharing IPv4 addresses among multiple CEs is susceptible to issues related to traffic logging. For details, see [RFC6269] sections 8 and 13.1.

3. Geo-location aware service

Sharing IPv4 addresses among multiple CEs is susceptible to issues related to geo-location. For details, see [RFC6269] section 7.

4. User Management

MAP IPv4 address assignment, and hence the IPv4 service itself, is tied to the IPv6 prefix lease; thus, the MAP service is also tied to this in terms of authorization, accounting, etc. For example, the MAP address has the same lifetime as its associated IPv6 prefix.

5. MAP Address Planning

This section is purposed to provide a referential guidance to operators, illustrating a common method of address planning with MAP in IPv4 residual deployment.

5.1. Planning for Residual Deployment, a Step-by-step Guide

Residual deployment starts from IPv6 address planning.

(A) IPv6 considerations

(A1) Determine the maximum number N of CEs to be supported, and, for generality, suppose N = 2^n.

For example, we suppose n = 20. It means there will be up to about one million CEs.

(A2) Choose the length x of IPv6 prefixes to be assigned to ordinary customers.

Consider we have a /32 IPv6 block, it is not a problem for the IPv6 deployment with the given number of CEs. Let x = 60, allowing subnets inside in each CE delegated networks.

(A3) Multiply N by a margin coefficient K, a power of two (K = 2 ^ k), to take into account that:

- Some privileged customers may be assigned IPv6 prefixes of length x', shorter than x, to have larger addressing spaces than ordinary customers, both in IPv6 and IPv4;

- Due to the hierarchy of routable prefixes, many theoretically delegable prefixes may not be actually delegable (ref: host density ratio of [RFC3194]).

In our example, let's take k = 0 for simplicity.

(B) IPv4 considerations

(B1) List all (non overlapping, not yet assigned to any in-running networks) IPv4 prefixes {Hi} that are available for IPv4 residual deployment.

Suppose that we hold two blocks and not yet assigned to any fixed network: 192.0.2.0/24 and 198.51.100.0/24.

(B2) Take enough of them, among the shortest ones, to get a total whose size M is a power of two (M = 2 ^ m), and includes a good proportion of the available IPv4 space.

If we use both blocks, M = 2^24 + 2^24, and therefore m = 25. Suppose the intended sharing ratio is 8 subscribers per address, resulting in (65536 - 1024)/8 = 8064 ports per subscriber assuming that the well-known ports are excluded. Then the PSID length to achieve this will be log2(8) = 3 bits. Bearing in mind the IPv4 24 bit prefix length for each of our two prefixes, the EA-bit length is (32 - 24) + 3 = 11 bits.

(B3) For each IPv4 prefix, Hi, of length hi, choose an prefix extension, say Ri of length ri = m - (32 - hi).

All these indexes must be non overlapping prefixes (e.g. 0, 10, 110, 111 for one /10, one /11, and two /12). In our example, we pick 0 for a contiguous address block while 1 for another.

Then we have:

H1 = 192.0.2.0/24, h1 = 24, r1 = 17 => R1 = bin(0);
H2 = 198.51.100.0/24, h2 = 24, r2 = 17 => R2 = bin(1);

Sometimes the IPv4 residual pool is not well aggregated and the contiguous address blocks may have different sizes. For example, in (B1), if we have H1 = 59.112.0.0/13 and H2 = 219.120.0.0/16 as the IPv4 residual pool, then M = 2^19 + 2^16, and in such a case, we must pick m so that m = ceil(log2(M)), where "ceil(x)" means the minimum integer not less than x, i.e., m = 20 in this case. Therefore r1 = 20 - (32 - 13) = 1, while r2 = 20 - (32 - 16) = 4. Several combinations are available for the R1 and R2 and one only needs to pay attention to avoiding overlapping when picking up the values.

(C) After (A) and (B), derive the rule(s)

(C1) Derive the length c of the MAP domain IPv6 prefix, C, that will appear at the beginning of all delegated prefixes (c = x - (n + k)).

(C2) Take any prefix for this C of length c that starts with a RIR-allocated IPv6 prefix.

(C3) For each IPv4 prefix Hi, make the rule, in which the key is Hi and the value is the domain IPv6 prefix C followed by the rule index Ri. Then this i-th rule’s Rule IPv6 Prefix will have the length of (c + ri).

Then we can do that:

```
c = 40 => C = 2001:0db8:ff00::/40
Rule 1: Rule IPv6 Prefix = 2001:0db8:ff00::/41
Rule 2: Rule IPv6 Prefix = 2001:0db8:ff80::/41
```

If we have different lengths for the Rule IPv4 prefix (as the extra example discussed at the end of (B)), their Rule IPv6 prefixes should not have the same length, as their rule index length is different.

As a result, for a certain CE delegating 2001:0db8:ff98:7650::/60, its parameters are:

```
Rule IPv6 Prefix = 2001:0db8:ff80::/41 => Rule 2
IPv4 Suffix = bin(111 0110 0)
                              PSID = bin(101) = 0x5
Rule IPv4 Prefix = 198.51.100.0/24
CE IPv4 Address = 198.51.100.236
```

If different sharing ratio is demanded, we may partition CEs into groups and do (A) and (B) for each group, determining the PSID length for them separately.

5.2. Remarks on Deployment Paradigms

1. IPv6 address planning in residual deployment is independent of the usage of the residual IPv4 addresses. The IPv4 address pool for "residual deployment" contains IPv4 addresses not yet allocated to customers/subscribers and/or those already recalled from ex-customers, re-programmed into relatively well-aggregated blocks.

2. It is recommended to have the number of rule entries as less as possible so that the merit of statelss deployment is reflected in practical performances. However, this effort is often constrained by the condition of an operator whether (a): it holds large-enough contiguous IPv4 address block(s) for the residual deployment, and (b): a short-enough IPv6 domain prefix so that the /64 delegation is easily satisfied even the EA-bits is quite long. When condition (a) is not satisfied, sub-domains have to be defined for each relatively small but contiguous aggregated block; when condition (b) is not satisfied, one has to divide the IPv4 aggregates into smaller blocks artificially in order to reduce the length of EA-bits. When we have good conditions fitting (a) and (b), it is NOT recommended to define short EA-bits with small length of IPv4 suffix (the value p) nor to increase the number of rule entries (also the number of sub-

domains) unless it really has to.

3. An extreme case is, when EA-bits contain the full IPv4 address while a full IPv4 address is assigned to a CE, i.e., o = p = 32, and q = 0, the MAP address format becomes almost equivalent to RFC6052-format [RFC6052] except the off-domain IPv4 peer's mapped IPv6 address. This frees the domain to distribute rules but the DMR. In such a case, IPv6 addressing is fully dependent of IPv4, which defers from the typical residual deployment case. MAP is mainly designed for residual deployment but also applied for the case of legacy IPv4 networks keeping communication with the IPv4 world over the IPv6 domain without renumbering, as long as the address planning doesn't matter.

4. Another extreme case is, when EA-bits' length becomes to zero, i.e., o = p = q = 0, a rule actually defines a correspondence between an IPv6 address and an IPv4 address (or a prefix), without any algorithmic correlation to each other. Using such a case in practice is not prohibited by the specification, but it is not recommended to deploy null EA-bits in large scale as the concern discussed in the above Remark 2, and as it has the limitation that the PSID must be null (q = 0) and therefore multiple CEs sharing a same IPv4 address is not supported here. It is recommended to apply Lightweight 4over6 [I-D.ietf-softwire-lw4over6], if a full de-correlation between IPv6 address and IPv4 address as well as port range is demanded.

5. A not-so-extreme case, p = 0, o = q, i.e., only PSID is applied for the EA-bits, is also a case possibly happening in practice. It also potentially generates a huge number of rules and therefore large-scale deployment of this case is not recommended either.

6. For operators who would like to utilize "some bits" of IPv6 address to do service identification, QoS differentiation, etc., it is recommended that these special-purpose bits should be embedded before the EA-bits so as to reduce the possibility of bit-conflict. However, it requires quite shorter IPv6 aggregate prefix of the operator. The bit-conflict is more likely to happen in this case if different domains have different Rule prefix lengths. Operators with this demand should pay attention to the impact on the domain rule planning.

6. Migration Methodology

6.1. Roadmap for MAP-based Solution

6.1.1. Start from Scratch

IPv6 deployment normally involves a step-wise approach where parts of the network should properly updated gradually. As IPv6 deployment progresses it may be simpler for operators to employ a single-version network, since deploying both IPv4 and IPv6 in parallel would cost more than IPv6-only network. Therefore switching to an IPv6-only network in relatively small scale will become more prevalent. Meanwhile, a significant part of network will still stay in IPv4 for long time, especially at early stage of IPv6 transition. There may not be enough public or private IPv4 addresses to support end-to-end network communication, without segmenting the network into small parts with sharing one IPv4 address space. That is a time to introduce MAP to bridge these IPv4 islands through IPv6 network.

6.1.2. Coexiting Phases

SP has various deployment strategy in the middle of transition. It’s foreseeable that IPv6 would likely coexist with IPv4 in a long period. The MAP deployment would also fit into the coexisting mode. To be specific, dual-stack technology is recommended in RFC6180 as the simplest deployment model to advance IPv6 deployment. MAP technology could get along well with native IPv6 connections and compatible with residual IPv4 networks. RFC6264 described a incremental transition approach in order to migrate networks to IPv6-only. DS-Lite is treated as a technology to accelerate the whole process. MAP can also take the same role to achieve a smooth transition.

6.1.3. Exit Strategy

The benefit of IPv6-only + MAP is that all IPv6 flows would go directly to the Internet, no need for encapsulation or translation. In this way, as more content providers and service are available over IPv6, the utilization on MAP CE and BR goes down since fewer destinations require MAP progressing. This way would advance IPv6, because it provides everyone incentives to use IPv6, and eventually the result is an pure IPv6 network with no need for IPv4. As more content providers and hosts equipped with IPv6 capabilities , the MAP utilization goes down until it is eventually not used at all when all content is IPv6. In this way, MAP has an "exit strategy". The corresponding solutions will leave the network in time.

6.2. Migration Mode

IPv4 Residual deployment is a interim phase during IPv6 migration. It would be beneficial to ISPs, if this phase is as short as possible since end-to-end IPv6 traversal is the really goals. When IPv6 is getting more and more mature, MAP would be retired in a natural way .

6.2.1. Passive Transition

Passive Transition is following IPv4 retirement law. In another word, MAP would always get along with IPv4, even all nodes is dual-stack capable. At a later stage of IPv6 migration, MAP can also be served for dual-stack hosts, which is sending traffic through the IPv4 stack. There is still a value for this approach because it could steer IPv4 traffic to IPv6 going through a MAP CE processing. When it comes the time ISP decide to turn off IPv4, MAP would be unnecessary due to IPv4 disappearance.

6.2.2. Active Transition

Active Transition is targeting to accelerate IPv4 exit and increase native IPv6 utilization. A desirable way deploying MAP is only providing IPv6 traversal ability to a IPv4-only host. However, MAP CE can not determine received traffic is send from a IPv4 node or a dual-stack node. In the latter case, IPv6 utilization is preferred for the most part . When a network evolves to a post-IPv6 era, it might be good for ISPs to consider to implement enforcement rules to help IPv6 migration.

- ISP could install only IPv6 record (i.e. AAAA) in DNS server, which would provide users with IPv6 steering effects. When a host is IPv6-capable and gets IPv6 DNS reply in advance, MAP functionalities would be restricted by IPv6-only record response.

- ISP could retrieve shared IPv4 address by increasing sharing ratio. In this case, number of concurrent IPv4 sessions on MAP CE would be suppressed. It would encourage native IPv6 growth in some extent.

- ISP could allocate a dedicated IPv6 prefix for MAP deployment. The allocation could not only facilitate the differentiation between MAPed traffic and native IPv6 traffic, but also clearly observe the change of MAP traffic. When the traffic is reducing for a while, ISP could close the MAP functionalities in some specific area. It would result networks to native IPv6-only capable.

7. IANA Considerations

This specification does not require any IANA actions.

8. Security Considerations

There are no new security considerations pertaining to this document.

9. Contributors

The members of the MAP design team are:

Congxiao Bao, Mohamed Boucadair, Gang Chen, Maoke Chen, Wojciech Dec, Xiaohong Deng, Remi Despres, Jouni Korhonen, Xing Li, Satoru Matsushima, Tomasz Mrugalski, Tetsuya Murakami, Jacni Qin, Qiong Sun, Tina Tsou, Dan Wing, Leaf Yeh, and Jan Zorz.

Thanks to Chunfa Sun who was an active co-author of some earlier versions of this draft. Thanks to Shishio Tsuchiya’s valueable suggestion for this document.

10. Acknowledgements

Remi Despres contributed the original example of step-by-step deployment guidance in discussion with the authors. Ole Troan, as the head of MAP Design Team, joined the discussion directly and contributed a lot of ideas and comments. We also thank other members of the MAP Design Team for their comments and suggestions.

Thanks to Tom Talyer, Qi Sun and Ian Farrer for their thorough review and helpful comments.

11. References

11.1. Normative References

[I-D.ietf-softwire-4rd]
Despres, R., Jiang, S., Penno, R., Lee, Y., Chen, G., and M. Chen, "IPv4 Residual Deployment via IPv6 - a Stateless Solution (4rd)", draft-ietf-softwire-4rd-10 (work in progress), December 2014.

[I-D.ietf-softwire-map]
Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, "Mapping of Address and Port with Encapsulation (MAP)", draft-ietf-softwire-map-13 (work in progress), March 2015.

[I-D.ietf-softwire-map-dhcp]
Mrugalski, T., Troan, O., Farrer, I., Perreault, S., Dec, W., Bao, C., Yeh, L., and X. Deng, "DHCPv6 Options for configuration of Softwire Address and Port Mapped Clients", draft-ietf-softwire-map-dhcp-12 (work in progress), March 2015.

[I-D.ietf-softwire-map-t]
Li, X., Bao, C., Dec, W., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", draft-ietf-softwire-map-t-08 (work in progress), December 2014.

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

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

[RFC6791] Li, X., Bao, C., Wing, D., Vaithianathan, R., and G. Huston, "Stateless Source Address Mapping for ICMPv6 Packets", RFC 6791, November 2012.

11.2. Informative References

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

[RFC3194] Durand, A. and C. Huitema, "The H-Density Ratio for Address Assignment Efficiency An Update on the H ratio", RFC 3194, November 2001.

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

[RFC7368] Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", RFC 7368, October 2014.

Authors' Addresses

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

Phone: +86 10 5855 2923
Email: sunqiong@ctbri.com.cn

Maoke Chen
BBIX, Inc.
Tokyo Shiodome Building, Higashi-Shimbashi 1-9-1
Minato-ku, Tokyo 105-7310
Japan

Email: maoke@bbix.net

Gang Chen
China Mobile
28 Xuanwumenxi Ave; Xuanwu District
Beijing
P.R. China

Email: chengang@chinamobile.com

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

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

Simon Perreault
Jive Communications
Quebec, QC
Canada

Email: sperreault@jive.com

Softwires Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 5, 2015

X. Li
C. Bao
CERNET Center/Tsinghua University
W. Dec, Ed.
O. Troan
Cisco Systems
S. Matsushima
SoftBank Telecom
T. Murakami
IP Infusion

December 2, 2014

Mapping of Address and Port using Translation (MAP-T)
draft-ietf-softwire-map-t-08

Abstract

This document specifies the "Mapping of Address and Port" stateless IPv6-IPv4 Network Address Translation (NAT64) based solution architecture for providing shared or non-shared IPv4 address connectivity to and across an IPv6 network.

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 June 5, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Conventions 3
3. Terminology 3
4. Architecture 5
5. Mapping Rules 7
5.1. Destinations outside the MAP domain 7
6. The IPv6 Interface Identifier 7
7. MAP-T Configuration 8
7.1. MAP CE 8
7.2. MAP BR 9
8. MAP-T Packet Forwarding 9
8.1. IPv4 to IPv6 at the CE 9
8.2. IPv6 to IPv4 at the CE 10
8.3. IPv6 to IPv4 at the BR 11
8.4. IPv4 to IPv6 at the BR 11
9. ICMP Handling 11
10. Fragmentation and Path MTU Discovery 12
10.1. Fragmentation in the MAP domain 12
10.2. Receiving IPv4 Fragments on the MAP domain borders . . . 12
10.3. Sending IPv4 fragments to the outside 13
11. NAT44 Considerations 13
12. Usage Considerations 13
12.1. EA-bit length 0 13
12.2. Mesh and Hub and spoke modes 13
12.3. Communication with IPv6 servers in the MAP-T domain . . 14
12.4. Compatibility with other NAT64 solutions 14
13. IANA Considerations 14
14. Security Considerations 14
15. Contributors 15
16. Acknowledgements 16
17. References 16
17.1. Normative References 16
17.2. Informative References 16
Appendix A. Examples of MAP-T translation 19
Appendix B. Port mapping algorithm 22
Authors' Addresses 22

1. Introduction

Experiences from initial service provider IPv6 network deployments, such as [RFC6219], indicate that successful transition to IPv6 can happen while supporting legacy IPv4 users without a full end-to-end dual IP stack deployment. However, due to public IPv4 address exhaustion this requires an IPv6 technology that supports IPv4 users utilizing shared IPv4 addressing, while also allowing the network operator to optimize their operations around IPv6 network practices. The use of double NAT64 translation based solutions is an optimal way to address these requirements, especially in combination with stateless translation techniques that minimize operational challenges outlined in [I-D.ietf-softwire-stateless-4v6-motivation].

The Mapping of Address and Port - Translation (MAP-T) architecture specified in this draft is such a double stateless NAT64 based solution. It builds on existing stateless NAT64 techniques specified in [RFC6145], along with the stateless algorithmic address & transport layer port mapping scheme defined in MAP-E [I-D.ietf-softwire-map]. The MAP-T solution differs from MAP-E in the use of IPv4-IPv6 translation, rather than encapsulation, as the form of IPv6 domain transport. The translation mode is considered advantageous in scenarios where the encapsulation overhead, or IPv6 operational practices (e.g. Use of IPv6 only servers, or reliance on IPv6 + protocol headers for traffic classification) rule out encapsulation. These scenarios are presented in [I-D.maglione-softwire-map-t-scenarios]

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Terminology

MAP-T
: Mapping of Address and Port by means of address Translation.

MAP Customer Edge (CE)
: A device functioning as a Customer Edge (CE) router in a MAP deployment. A typical MAP CE adopting MAP rules will serve a residential site with one WAN side IPv6 addressed interface, and one or more LAN side interfaces addressed using private IPv4 addressing.

MAP Border Relay (BR):	A MAP enabled router managed by the service provider at the edge of a MAP domain. A Border Relay (BR) router has at least an IPv6-enabled interface and an IPv4 interface connected to the native IPv4 network. A MAP BR may also be referred to simply as a "BR" within the context of MAP.
MAP domain:	One or more MAP CEs and BRs connected by means of an IPv6 network and sharing a common set of MAP Rules. A service provider may deploy a single MAP domain, or may utilize multiple MAP domains.
MAP Rule:	A set of parameters describing the mapping between an IPv4 prefix, IPv4 address or shared IPv4 address and an IPv6 prefix or address. Each MAP domain uses a different mapping rule set.
MAP Rule set:	A Rule set is composed out of all the MAP Rules communicated to a device, that are intended for determining the devices' IP+port mapping and forwarding operations. The MAP Rule set is interchangeably referred to in this document as a MAP Rule table or simply Rule table. Two specific types of rules, Basic Mapping Rule (BMR) and Forward Mapping Rule (FMR), are defined in Section 5 of [I-D.ietf-softwire-map]. The Default Mapping Rule (DMR) is defined in this document.
MAP Rule table:	See MAP Rule set.
MAP node:	A device that implements MAP.
Port-set:	Each node has a separate part of the transport layer port space; denoted as a port-set.
Port-set ID (PSID):	Algorithmically identifies a set of ports exclusively assigned to the CE.
Shared IPv4 address:	An IPv4 address that is shared among multiple CEs. Only ports that belong to the assigned port-set can be used for communication. Also known as a Port-Restricted IPv4 address.

End-user IPv6 prefix: The IPv6 prefix assigned to an End-user CE by other means than MAP itself. E.g. Provisioned using DHCPv6 PD [RFC3633], assigned via SLAAC [RFC4862], or configured manually. It is unique for each CE.

MAP IPv6 address: The IPv6 address used to reach the MAP function of a CE from other CEs and from BRs.

Rule IPv6 prefix: An IPv6 prefix assigned by a Service Provider for a MAP rule.

Rule IPv4 prefix: An IPv4 prefix assigned by a Service Provider for a MAP rule.

Embedded Address (EA) bits: The IPv4 EA-bits in the IPv6 address identify an IPv4 prefix/address (or part thereof) or a shared IPv4 address (or part thereof) and a port-set identifier.

4. Architecture

Figure 1 depicts the overall MAP-T architecture, which sees any number of privately addressed IPv4 users (N and M) connected by means of MAP-T CEs to an IPv6 network that is equipped with one or more MAP-T BR. CEs and BRs that share MAP configuration parameters, referred to as MAP rules, form a MAP-T Domain.

Functionally the MAP-T CE and BR utilize and extend some well established technology building blocks to allow the IPv4 users to correspond with nodes on the Public IPv4 network, or IPv6 network as follows:

- A (NAT44) NAPT [RFC2663] function on a MAP CE is extended with support for restricting the allowable TCP/UDP ports for a given IPv4 address. The IPv4 address and port range used are determined by the MAP provisioning process and identical to MAP-E [I-D.ietf-softwire-map].

- A stateless NAT64 function [RFC6145] is extended to allow stateless mapping of IPv4 and transport layer port ranges to IPv6 address space.

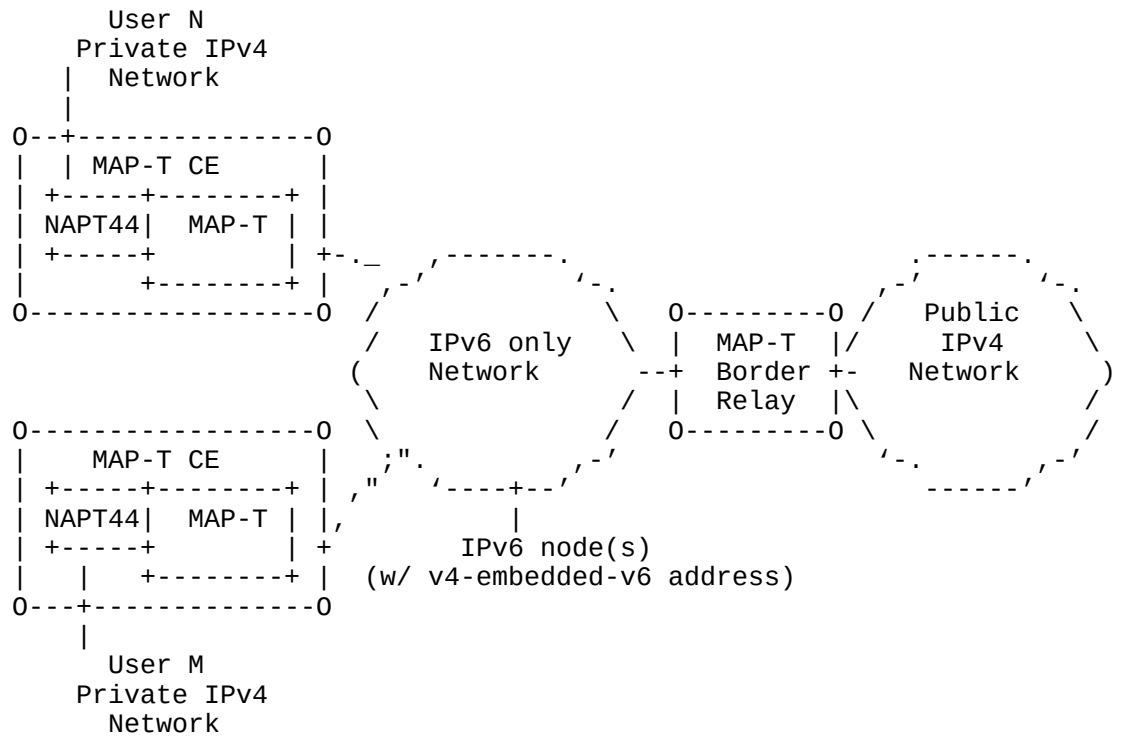


Figure 1: MAP-T Architecture

Each MAP-T CE is assigned with a regular IPv6 prefix from the operator's IPv6 network. This, in conjunction with MAP domain configuration settings and the use of the MAP procedures allows the computation of a MAP IPv6 address and a corresponding IPv4 address. To allow for IPv4 address sharing, the CE may also have be configured with a TCP/UDP port-range that is identified by means of a MAP Port Set Identifier (PSID) value. Each CE is responsible for forwarding traffic between a given user's private IPv4 address space and the MAP domain's IPv6 address space. The IPv4-IPv6 adaptation uses stateless NAT64, in conjunction with the MAP algorithm for address computation.

The MAP-T BR connects one or more MAP-T domains to external IPv4 networks using stateless NAT64 as extended by the MAP-T behaviour described in this document.

In contrast to MAP-E, NAT64 technology is used in the architecture for two purposes. Firstly, it is intended to diminish encapsulation overhead and allow IPv4 and IPv6 traffic to be treated as similarly as possible. Secondly, it is intended to allow IPv4-only nodes to

correspond directly with IPv6 nodes in the MAP-T domain that have IPv4 embedded IPv6 addresses as per [RFC6052]).

The MAP-T architecture is based on the following key properties i) algorithmic IPv4-IPv6 address mapping codified as MAP Rules covered in Section 5 ii) A MAP IPv6 address identifier, described in Section 6 iii) MAP-T IPv4-IPv6 forwarding behavior described in Section 8.

5. Mapping Rules

The MAP-T algorithmic mapping rules are identical to those in Section 5 of the MAP-E specification [I-D.ietf-softwire-map], with the following exception. The forwarding of traffic to and from IPv4 destinations outside a MAP-T domain is to be performed as described here under, instead of Section 5.4 of the MAP-E specification.

5.1. Destinations outside the MAP domain

IPv4 traffic sent by MAP nodes that are all within one MAP domain is translated to IPv6, with the sender’s MAP IPv6 address, derived via the Basic Mapping Rule (BMR), as the IPv6 source address and the recipient’s MAP IPv6 address, derived via the Forward Mapping Rule (FMR), as the IPv6 destination address.

IPv4 addressed destinations outside of the MAP domain are represented by means of IPv4-Embedded IPv6 address as per [RFC6052], using the BR’s IPv6 prefix. For a CE sending traffic to any such destination, the source address of the IPv6 packet will be that of the CE’s MAP IPv6 address, and the destination IPv6 address will be the destination IPv4-embedded-IPv6 address. This address mapping is termed as following the MAP-T Default Mapping Rule (DMR) and is defined in terms of the IPv6 prefix advertised by one or more BRs, which provide external connectivity. A typical MAP-T CE will install an IPv4 default route using this rule. A BR will use this rule when translating all outside IPv4 source addresses to the IPv6 MAP domain.

The DMR IPv6 prefix-length SHOULD be by default 64 bits long, and in any case MUST NOT exceed 96 bits. The mapping of the IPv4 destination behind the IPv6 prefix will by default follow the /64 rule as per [RFC6052]. Any trailing bits after the IPv4 address are set to 0x0.

6. The IPv6 Interface Identifier

The Interface identifier format of a MAP-T node is the same as described in section 6 of [I-D.ietf-softwire-map]. For convenience this is cited below:

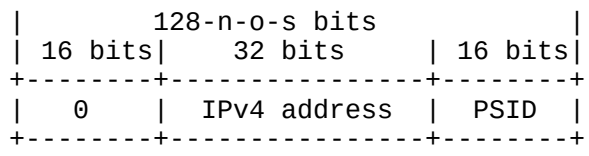


Figure 2

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

If the End-user IPv6 prefix length is larger than 64, the most significant parts of the interface identifier is overwritten by the prefix.

7. MAP-T Configuration

For a given MAP domain, the BR and CE MUST be configured with the following MAP parameters. The values for these parameters are identical for all CEs and BRs within a given MAP-T domain.

- The Basic Mapping Rule and optionally the Forwarding Mapping Rules, including the Rule IPv6 prefix, Rule IPv4 prefix, and Length of Embedded Address bits
- Use of Hub and spoke mode or Mesh mode. (If all traffic should be sent to the BR, or if direct CE to CE correspondence should be supported).
- Use of IPv4-IPv6 Translation (MAP-T)
- The BR’s IPv6 prefix used in the DMR

7.1. MAP CE

For a given MAP domain, the MAP configuration parameters are the same across all CEs within that domain. These values may be conveyed and configured on the CEs using a variety of methods, including; DHCPv6, Broadband Forum’s "TR-69" Residential Gateway management interface, Netconf, or manual configuration. This document does not prescribe any of these methods, but recommends that a MAP CE SHOULD implement DHCPv6 options as per [I-D.ietf-softwire-map-dhcp]. Other configuration and management methods may use the data model described by this option for consistency and convenience of implementation on CEs that support multiple configuration methods.

Besides the MAP configuration parameters, a CE requires an IPv6 prefix to be assigned to the CE. This End-user IPv6 prefix is configured as part of obtaining IPv6 Internet access, and is acquired using standard IPv6 means applicable in the network where the CE is located.

The MAP provisioning parameters, and hence the IPv4 service itself, are tied to the End-user IPv6 prefix; thus, the MAP service is also tied to this in terms of authorization, accounting, etc.

A single MAP CE MAY be connected to more than one MAP domain, just as any router may have more than one IPv4-enabled service provider facing interface and more than one set of associated addresses assigned by DHCPv6. Each domain a given CE operates within would require its own set of MAP configuration elements and would generate its own IPv4 address. Each MAP domain requires a distinct End-user IPv6 prefix.

7.2. MAP BR

The MAP BR MUST be configured with the same MAP elements as the MAP CEs operating within the same domain.

For increased reliability and load balancing, the BR IPv6 prefix MAY be shared across a given MAP domain. As MAP is stateless, any BR may be used for forwarding to/from the domain at any time.

Since MAP uses provider address space, no specific IPv6 or IPv4 routes need to be advertised externally outside the service provider’s network for MAP to operate. However, the BR prefix needs to be advertised in the service provider’s IGP.

8. MAP-T Packet Forwarding

The end-to-end packet flow in MAP-T involves an IPv4 or IPv6 packet being forwarded by a CE of BR in one of two directions for each such case. This section presents a conceptual view of the operations involved in such forwarding.

8.1. IPv4 to IPv6 at the CE

A MAP-T CE receiving IPv4 packets SHOULD perform NAPT NAT44 processing, and create any necessary NAPT44 bindings. The source address and source port-range of packets resulting from the NAPT44 processing MUST correspond to the source IPv4 address and source transport port-range assigned to the CE by means of the MAP Basic Mapping Rule (BMR).

The IPv4 packet is subject to a longest IPv4 destination address + port match MAP rule selection, which then determines the parameters for the subsequent NAT64 operation. By default, all traffic is matched to the default mapping rule (DMR), and subject to the stateless NAT64 operation using the DMR parameters for NAT64 Section 5.1. Packets that are matched to (optional) Forward Mapping Rules (FMRs) are subject to the stateless NAT64 operation using the FMR parameters Section 5 for the MAP algorithm. In all cases the CE's MAP IPv6 address Section 6 is used as a source address.

A MAP-T CE MUST support a Default Mapping Rule and SHOULD support one or more Forward Mapping Rules.

8.2. IPv6 to IPv4 at the CE

A MAP-T CE receiving an IPv6 packet performs its regular IPv6 operations (filtering, pre-routing, etc). Only packets that are addressed to the CE's MAP-T IPv6 addresses, and with source addresses matching the IPv6 map-rule prefixes of a DMR or FMR, are processed by the MAP-T CE, with the DMR or FMR being selected based on a longest match. The CE MUST check that each MAP-T received packet's destination transport-layer destination port number is in the range allowed for by the CE's MAP BMR configuration. The CE MUST silently drop any non conforming packet and an appropriate counter incremented. When receiving a packet whose source IP address longest matches an FMR prefix, the CE MUST perform a check of consistency of the source address against the allowed values as per the derived allocated source port-range. If the source port number of a packet is found to be outside the allocated range, the CE MUST drop the packet and SHOULD respond with an ICMPv6 "Destination Unreachable, Source address failed ingress/egress policy" (Type 1, Code 5).

For each MAP-T processed packet, the CE's NAT64 function MUST compute an IPv4 source and destination addresses. The IPv4 destination address is computed by extracting relevant information from the IPv6 destination and the information stored in the BMR as per Section 5. The IPv4 source address is formed by classifying a packet's source as longest matching a DMR or FMR rule prefix, and then using the respective rule parameters for the NAT64 operation.

The resulting IPv4 packet is then forwarded to the CE's NAPT NAPT44 function, where the destination IPv4 address and port number MUST be mapped to their original value, before being forwarded according to the CE's regular IPv4 rules. When the NAPT44 function is not enabled, by virtue of MAP configuration, the traffic from the stateless NAT64 function is directly forwarded according to the CE's IPv4 rules.

8.3. IPv6 to IPv4 at the BR

A MAP-T BR receiving an IPv6 packet MUST select a matching MAP rule based on a longest address match of the packet's source address against the MAP Rules present on the BR. In combination with the Port-Set-Id derived from the packet's source IPv6 address, the selected MAP rule allows the BR to verify that the CE is using its allowed address and port range. Thus, the BR MUST perform a validation of the consistency of the source against the allowed values from the identified port-range. If the packet's source port number is found to be outside the range allowed, the BR MUST drop the packet and increment a counter to indicate the event. The BR SHOULD also respond with an ICMPv6 "Destination Unreachable, Source address failed ingress/egress policy" (Type 1, Code 5).

When constructing the IPv4 packet, the BR MUST derive the source and destination IPv4 addresses as per Section 5 of this document and translate the IPv6 to IPv4 headers as per [RFC6145]. The resulting IPv4 packet is then passed to regular IPv4 forwarding.

8.4. IPv4 to IPv6 at the BR

A MAP-T BR receiving IPv4 packets uses a longest match IPv4 + transport layer port lookup to identify the target MAP-T domain and select the FMR and DMR rules. The MAP-T BR MUST then compute and apply the IPv6 destination addresses from the IPv4 destination address and port as per the selected FMR. The MAP-T BR MUST also compute and apply the IPv6 source addresses from the IPv4 source address as per Section 5.1 (i.e. Using the IPv4 source and the BR's IPv6 prefix it forms an IPv6 embedded IPv4 address). Throughout the generic IPv4 to IPv6 header translation procedures following [RFC6145] apply. The resulting IPv6 packets are then passed to regular IPv6 forwarding.

Note that the operation of a BR when forwarding to/from MAP-T domains that are defined without IPv4 address sharing is the same as that of stateless NAT64 IPv4/IPv6 translation.

9. ICMP Handling

MAP-T CEs and BRs MUST follow ICMP/ICMPv6 translation as per [RFC6145], however additional behavior is also required due to the presence of NAPT44. Unlike TCP and UDP, which provide two transport protocol port fields to represent both source and destination, the ICMP/ICMPv6 [RFC0792], [RFC4443] Query message header has only one ID field which needs to be used to identify a sending IPv4 host. When receiving IPv4 ICMP messages, the MAP-T CE MUST rewrite the ID field to a port value derived from the CE's Port-Set-Id.

A MAP-T BR receiving an IPv4 ICMP packet , which contains an ID field that is bound for a shared address in the MAP-T domain, SHOULD use the ID value as a substitute for the destination port in determining the IPv6 destination address. In all other cases, the MAP-T BR MUST derive the destination IPv6 address by simply mapping the destination IPv4 address without additional port info.

10. Fragmentation and Path MTU Discovery

Due to the different sizes of the IPv4 and IPv6 header, handling the maximum packet size is relevant for the operation of any system connecting the two address families. There are three mechanisms to handle this issue: Path MTU discovery (PMTUD), fragmentation, and transport-layer negotiation such as the TCP Maximum Segment Size (MSS) option [RFC0897]. MAP can use all three mechanisms to deal with different cases.

Note: The NAT64 [RFC6145]mechanism is not lossless. When IPv4 originated communication traverses across a double NAT64 function (a.k.a. NAT464), any IPv4 originated ICMP-independent PathMTU Discovery, as specified in [RFC 4821], ceases to be entirely reliable. This is because the[RFC4821] defined DF=1/MF=1 combination, following a double NAT64 translation, results in DF=0/ MF=1.

10.1. Fragmentation in the MAP domain

Translating an IPv4 packet to carry it across the MAP domain will increase its size typically by 20 bytes. The MTU in the MAP domain should be well managed and the IPv6 MTU on the CE WAN side interface SHOULD be configured so that no fragmentation occurs within the boundary of the MAP domain.

Fragmentation in MAP-T domain SHOULD be handled as described in section 4 and 5 of [RFC6145].

10.2. Receiving IPv4 Fragments on the MAP domain borders

Forwarding of an IPv4 packet received from the outside of the MAP domain requires the IPv4 destination address and the transport protocol destination port. The transport protocol information is only available in the first fragment received. As described in section 5.3.3 of [RFC6346] a MAP node receiving an IPv4 fragmented packet from outside SHOULD reassemble the packet before sending the packet onto the MAP domain. If the first packet received contains the transport protocol information, it is possible to optimize this behavior by using a cache and forwarding the fragments unchanged. A

description of such a caching algorithm is outside the scope of this document.

10.3. Sending IPv4 fragments to the outside

Two IPv4 hosts behind two different MAP CE's with the same IPv4 address sending fragments to an IPv4 destination host outside the domain may happen to use the same IPv4 fragmentation identifier, resulting in incorrect reassembly of the fragments at the destination host. Given that the IPv4 fragmentation identifier is a 16 bit field, it can be used similarly to port ranges. Thus, a MAP CE SHOULD rewrite the IPv4 fragmentation identifier to a value equivalent to a port of its allocated port-set.

11. NAT44 Considerations

The NAT44 implemented in the MAP CE SHOULD conform with the behavior and best current practice documented in [RFC4787], [RFC5508], and [RFC5382]. In MAP address sharing mode (determined by the MAP domain /rule configuration parameters) the operation of the NAT44 MUST be restricted to the available port numbers derived via the basic mapping rule.

12. Usage Considerations

12.1. EA-bit length 0

The MAP solution supports use and configuration of domains where a BMR expresses an EA-bit length of 0. This results in independence between the IPv6 prefix assigned to the CE and the IPv4 address and/ or port-range used by MAP. The k-bits of PSID information may in this case be derived from the BMR.

The constraint imposed is that each such MAP domain be composed of just 1 MAP CE which has a predetermined IPv6 end-user prefix. The BR would be configured with an FMR for each such CPE, where the rule would uniquely associate the IPv4 address + optional PSID and the IPv6 prefix of that given CE.

12.2. Mesh and Hub and spoke modes

The hub and spoke mode of communication, whereby all traffic sent by a MAP-T CE is forwarded via a BR, and the mesh mode, whereby a CE is directly able to forward traffic to another CE, are governed by the activation of Forward Mapping Rule that cover the IPv4-prefix destination, and port-index range. By default, a MAP CE configured only with a BMR, as per this specification, will use it to configure its IPv4 parameters and IPv6 MAP address without enabling mesh mode.

12.3. Communication with IPv6 servers in the MAP-T domain

By default, MAP-T allows communication between both IPv4-only and any IPv6 enabled devices, as well as with native IPv6-only servers provided that the servers are configured with an IPv4-mapped IPv6 address. This address could be part of the IPv6 prefix used by the DMR in the MAP-T domain. Such IPv6 servers (e.g. An HTTP server, or a web content cache device) are thus able to serve both IPv6 users as well as IPv4-only users alike utilizing IPv6. Any such IPv6-only servers SHOULD have both A and AAAA records in DNS. DNS64 [RFC6147] become required only when IPv6 servers in the MAP-T domain are expected themselves to initiate communication to external IPv4-only hosts.

12.4. Compatibility with other NAT64 solutions

The MAP-T CEs NAT64 function is by default compatible for use with [RFC6146] stateful NAT64 devices that are placed in the operator's network. In such a case the MAP-T CE's DMR prefix is configured to correspond to the NAT64 device prefix. This in effect allows the use of MAP-T CEs in environments that need to perform statistical multiplexing of IPv4 addresses, while utilizing stateful NAT64 devices, and can take the role of a CLAT as defined in [RFC6877].

13. IANA Considerations

This specification does not require any IANA actions.

14. Security Considerations

Spoofing attacks: With consistency checks between IPv4 and IPv6 sources that are performed on IPv4/IPv6 packets received by MAP nodes, MAP does not introduce any new opportunity for spoofing attacks that would not already exist in IPv6.

Denial-of-service attacks: In MAP domains where IPv4 addresses are shared, the fact that IPv4 datagram reassembly may be necessary introduces an opportunity for DOS attacks. This is inherent to address sharing, and is common with other address sharing approaches such as DS-Lite and NAT64/DNS64. The best protection against such attacks is to accelerate IPv6 support in both clients and servers.

Routing-loop attacks: This attack may exist in some automatic tunneling scenarios are documented in [RFC6324]. They cannot exist with MAP because each BRs checks that the IPv6 source address of a received IPv6 packet is a CE address based on Forwarding Mapping Rule.

Attacks facilitated by restricted port-set: From hosts that are not subject to ingress filtering of [RFC2827], some attacks are possible by an attacker injecting spoofed packets during ongoing transport connections ([RFC4953], [RFC5961], [RFC6056]. The attacks depend on guessing which ports are currently used by target hosts, and using an unrestricted port-set is preferable, i.e. Using native IPv6 connections that are not subject to MAP port-range restrictions. To minimize this type of attacks when using a restricted port set, the MAP CE's NAT44 filtering behavior SHOULD be "Address-Dependent Filtering". Furthermore, the MAP CEs SHOULD use a DNS transport proxy function to handle DNS traffic, and source such traffic from IPv6 interfaces not assigned to MAP-T. Practicalities of these methods are discussed in Section 5.9 of [I-D.dec-stateless-4v6].

ICMP Flooding Given the necessity to process and translate ICMP and ICMPv6 messages by the BR and CE nodes, a foreseeable attack vector is that of a flood of such messages leading to a saturation of the node's ICMP computing resources. This attack vector is not specific to MAP, and its mitigation lies a combination of policing the rate of ICMP messages, policing the rate at which such messages can get processed by the MAP nodes, and of course identifying and blocking off the source(s) of such traffic.

[RFC6269] outlines general issues with IPv4 address sharing.

15. Contributors

The following individuals authored major contributions to this document, and made the document possible:

Chongfeng Xie (China Telecom) Room 708, No.118, Xizhimennei Street Beijing 100035 CN Phone: +86-10-58552116 Email: xiechf@ctbri.com.cn

Qiong Sun (China Telecom) Room 708, No.118, Xizhimennei Street Beijing 100035 CN Phone: +86-10-58552936 Email: sunqiong@ctbri.com.cn

Rajiv Asati (Cisco Systems) 7025-6 Kit Creek Road Research Triangle Park NC 27709 USA Email: rajiva@cisco.comc

Gang Chen (China Mobile) 53A,Xibianmennei Ave. Beijing 100053 P.R.China Email: chengang@chinamobile.com

Wentao Shang (CERNET Center/Tsinghua University) Room 225, Main Building, Tsinghua University Beijing 100084 CN Email: wentaoshang@gmail.com

Guoliang Han (CERNET Center/Tsinghua University) Room 225, Main Building, Tsinghua University Beijing 100084 CN Email: bupthgl@gmail.com

Yu Zhai CERNET Center/Tsinghua University Room 225, Main Building, Tsinghua University Beijing 100084 CN Email: jacky.zhai@gmail.com

16. Acknowledgements

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

The authors would also like to thank Mohamed Boucadair, Guillaume Gottard, Dan Wing, Jan Zorz, Nejc Scoberne, Tina Tsou, Gang Chen, Maoke Chen, Xiaohong Deng, Jouni Korhonen, Tomasz Mrugalski, Jacni Qin, Chunfa Sun, Qiong Sun, Leaf Yeh, Andrew Yourtchenko, Roberta Maglione and Hongyu Chen for their review and comments.

17. References

17.1. Normative References

[I-D.ietf-softwire-map] Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, "Mapping of Address and Port with Encapsulation (MAP)", draft-ietf-softwire-map-12 (work in progress), November 2014.

[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.

[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.

17.2. Informative References

[I-D.dec-stateless-4v6] Dec, W., Asati, R., and H. Deng, "Stateless 4Via6 Address Sharing", draft-dec-stateless-4v6-04 (work in progress), October 2011.

[I-D.ietf-softwire-map-dhcp]
Mrugalski, T., Troan, O., Farrer, I., Perreault, S., Dec, W., Bao, C., leaf.yeh.sdo@gmail.com, l., and X. Deng, "DHCPv6 Options for configuration of Softwire Address and Port Mapped Clients", draft-ietf-softwire-map-dhcp-11 (work in progress), November 2014.

[I-D.ietf-softwire-stateless-4v6-motivation]
Boucadair, M., Matsushima, S., Lee, Y., Bonness, O., Borges, I., and G. Chen, "Motivations for Carrier-side Stateless IPv4 over IPv6 Migration Solutions", draft-ietf-softwire-stateless-4v6-motivation-05 (work in progress), November 2012.

[I-D.maglione-softwire-map-t-scenarios]
Maglione, R., Dec, W., Leung, I., and E. Mallette, "Use cases for MAP-T", draft-maglione-softwire-map-t-scenarios-05 (work in progress), October 2014.

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

[RFC0897] Postel, J., "Domain name system implementation schedule", RFC 897, February 1984.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

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

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[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.

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

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

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

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

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

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

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

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

[RFC6219] Li, X., Bao, C., Chen, M., Zhang, H., and J. Wu, "The China Education and Research Network (CERNET) IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition", RFC 6219, May 2011.

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

[RFC6324] Nakibly, G. and F. Templin, "Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", RFC 6324, August 2011.

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, April 2013.

Appendix A. Examples of MAP-T translation

Example 1 - Basic Mapping Rule:

Given the following MAP domain information and IPv6 end-user prefix assigned to a MAP CE:

```
End-user IPv6 prefix:  2001:db8:0012:3400::/56
Basic Mapping Rule:    {2001:db8:0000::/40 (Rule IPv6 prefix),
                       192.0.2.0/24 (Rule IPv4 prefix),
                       16 (Rule EA-bits length)}
PSID length:           (16 - (32 - 24) = 8. (Sharing ratio of 256)
PSID offset:           6 (default)
```

A MAP node (CE or BR) can via the BMR, or equivalent FMR, determine the IPv4 address and port-set as shown below:

```
EA bits offset:        40
IPv4 suffix bits (p):  Length of IPv4 address (32) - IPv4 prefix
                       length (24) = 8
IPv4 address:          192.0.2.18 (0xc0000212)
PSID start:            40 + p = 40 + 8 = 48
PSID length (q):       o - p = (End-user prefix len -
                       rule IPv6 prefix len) - p
                       = (56 - 40) - 8 = 8
PSID:                  0x34
```

```
Available ports (63 ranges): 1232-1235, 2256-2259, ..... ,
                                  63696-63699, 64720-64723
```

The BMR information allows a MAP CE to determine (complete) its IPv6 address within the indicated end-user IPv6 prefix.

```
IPv6 address of MAP CE:  2001:db8:0012:3400:0000:c000:0212:0034
```

Example 2 - BR:

Another example can be made of a MAP-T BR,
configured with the following FMR when receiving a packet
with the following characteristics:

```
IPv4 source address:       10.2.3.4 (0x0a020304)
TCP source port:           80
IPv4 destination address:  192.0.2.18 (0xc0000212)
TCP destination port:      1232

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

MAP-T BR Prefix (DMR):    2001:db8:ffff::/64
```

The above information allows the BR to derive as follows
the mapped destination IPv6 address for the corresponding
MAP-T CE, and also the source IPv6 address for
the mapped IPv4 source address.

```
IPv4 suffix bits (p):  32 - 24 = 8 (18 (0x12))
PSID length:           8
PSID:  0               x34 (1232)
```

The resulting IPv6 packet will have the following header fields:

```
IPv6 source address:      2001:db8:ffff:0:000a:0203:0400::
IPv6 destination address: 2001:db8:0012:3400:0000:c000:0212:0034
TCP source Port:          80
TCP destination Port:     1232
```

Example 3- FMR:

An IPv4 host behind a MAP-T CE (configured as per the previous examples) corresponding with an IPv4 host 10.2.3.4 will have its packets converted into IPv6 using the DMR configured on the MAP-T CE as follows:

```
Default Mapping Rule:          {2001:db8:ffff::/64 (Rule IPv6 prefix),
                                0.0.0.0/0 (Rule IPv4 prefix)}

IPv4 source address:           192.0.2.18
IPv4 destination address:      10.2.3.4
IPv4 source port:              1232
IPv4 destination port:         80
MAP-T CE IPv6 source address:  2001:db8:0012:3400:0000:c000:0212:0034
IPv6 destination address:      2001:db8:ffff:0:000a:0203:0400::
```

Example 4 - Rule with no embedded address bits and no address sharing

```
End-user IPv6 prefix:     2001:db8:0012:3400::/56
Basic Mapping Rule:       {2001:db8:0012:3400::/56 (Rule IPv6 prefix),
                          192.0.2.1/32 (Rule IPv4 prefix),
                          0 (Rule EA-bits length)}
PSID length:              0 (Sharing ratio is 1)
PSID offset:              n/a
```

A MAP node can via the BMR or equivalent FMR, determine the IPv4 address and port-set as shown below:

```
EA bits offset:           0
IPv4 suffix bits (p):     Length of IPv4 address - IPv4 prefix
                          length = 32 - 32 = 0
IPv4 address:             192.0.2.18 (0xc0000212)
PSID start:               0
PSID length:              0
PSID:                     null
```

The BMR information allows a MAP CE also to determine (complete) its full IPv6 address by combining the IPv6 prefix with the MAP interface identifier (that embeds the IPv4 address).

```
IPv6 address of MAP CE:  2001:db8:0012:3400:0000:c000:0201:0000
```

Example 5 - Rule with no embedded address bits and address sharing (sharing ratio 256)

```
End-user IPv6 prefix:    2001:db8:0012:3400::/56
Basic Mapping Rule:      {2001:db8:0012:3400::/56 (Rule IPv6 prefix),
                         192.0.2.18/32 (Rule IPv4 prefix),
                         0 (Rule EA-bits length)}
PSID length:             (16 - (32 - 24)) = 8. Sharing ratio of 256.
                         Provisioned with DHCPv6.
PSID offset:             6 (default)
PSID:                    0x20 (Provisioned with DHCPv6)
```

A MAP node can via the BMR determine the IPv4 address and port-set as shown below:

```
EA bits offset:          0
IPv4 suffix bits (p):    Length of IPv4 address - IPv4 prefix
                         length = 32 -32 = 0
IPv4 address             192.0.2.18 (0xc0000212)
PSID start:              0
PSID length:             8
PSID:                    0x34

Available ports (63 ranges) : 1232-1235, 2256-2259, ..... ,
                              63696-63699, 64720-64723
```

The BMR information allows a MAP CE also to determine (complete) its full IPv6 address by combining the IPv6 prefix with the MAP interface identifier (that embeds the IPv4 address and PSID).

```
IPv6 address of MAP CE:  2001:db8:0012:3400:0000:c000:0212:0034
```

Note that the IPv4 address and PSID is not derived from the IPv6 prefix assigned to the CE, but provisioned separately using for example MAP options in DHCPv6.

Appendix B. Port mapping algorithm

The driving principles and the mathematical expression of the mapping algorithm used by MAP can be found in Appendix B of [I-D.ietf-softwire-map]

Authors' Addresses

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

Email: xing@cernet.edu.cn

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

Email: congxiao@cernet.edu.cn

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

Email: wdec@cisco.com

Ole Troan
Cisco Systems
Oslo
Norway

Email: ot@cisco.com

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

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

Tetsuya Murakami
IP Infusion
1188 East Arques Avenue
Sunnyvale
USA

Email: tetsuya@ipinfusion.com

Softwire
Internet-Draft
Intended status: Standards Track
Expires: June 21, 2016

Y. Cui
J. Dong
P. Wu
M. Xu
Tsinghua University
A. Yla-Jaaski
Aalto University

December 19, 2015

Softwire Mesh Management Information Base (MIB)
draft-ietf-softwire-mesh-mib-14

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 a softwire 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 June 21, 2016.

Copyright Notice

Copyright (c) 2015 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 2
2. The Internet-Standard Management Framework 2
3. Terminology 3
4. Structure of the MIB Module 3
4.1. The swmSupportedTunnelTable Subtree 3
4.2. The swmEncapsTable Subtree 3
4.3. The swmBGPNeighborTable Subtree 4
4.4. The swmConformance Subtree 4
5. Relationship to Other MIB Modules 4
5.1. Relationship to the IF-MIB 4
5.2. Relationship to the IP Tunnel MIB 5
5.3. MIB modules required for IMPORTS 5
6. Definitions 5
7. Security Considerations 13
8. IANA Considerations 14
9. Acknowledgements 14
10. References 14
10.1. Normative References 14
10.2. Informative References 16
Authors' Addresses 16

1. Introduction

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

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

2. The Internet-Standard Management Framework

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

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). They

are defined using the mechanisms stated in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2 (Structure of Management Information Version 2), which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

3. Terminology

This document uses terminology from the softwire problem statement RFC 4925 [RFC4925], the BGP encapsulation subsequent address family identifier (SAFI) and the BGP tunnel encapsulation attribute RFC 5512 [RFC5512], the softwire mesh framework RFC 5565 [RFC5565] and the BGP IPsec tunnel encapsulation attributeand RFC 5566 [RFC5566].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

4. Structure of the MIB Module

The softwire mesh MIB provides a method to monitor the softwire mesh objects through SNMP.

4.1. The swmSupportedTunnelTable Subtree

The swmSupportedTunnelTable subtree provides the information about what types of tunnels can be used for softwire mesh scenarios in the AFBR. The softwire mesh framework RFC 5565 [RFC5565] does not mandate the use of any particular tunneling technology. Based on the BGP tunnel encapsulation attribute tunnel types introduced by RFC 5512[RFC5512] and RFC 5566[RFC5566], the softwire mesh tunnel types include at least L2TPv3 (Layer Two Tunneling Protocol-Version 3) over IP, GRE (Generic Routing Encapsulation), Transmit tunnel endpoint, IPsec in Tunnel-mode, IP in IP tunnel with IPsec Transport Mode, MPLS-in-IP tunnel with IPsec Transport Mode and IP in IP. The detailed encapsulation information of different tunnel types (e.g., L2TPv3 Session ID, GRE Key, etc.) is not managed in the swmMIB.

4.2. The swmEncapsTable Subtree

The swmEncapsTable subtree provides softwire mesh NLRI-NH information (Network Layer Reachability Information-Next Hop) about the AFBR. It keeps the mapping between the External-IP (E-IP) prefix and the Internal-IP (I-IP) address of the next hop. The mappings determine which I-IP destination address will be used to encapsulate the received packet according to its E-IP destination address. The definitions of E-IP and I-IP are explained in section 4.1 of RFC

5565[RFC5565]. The number of entries in swmEncapsTable shows how many softwire mesh tunnels are maintained in this AFBR.

4.3. The swmBGPNeighborTable Subtree

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

4.4. The swmConformance Subtree

The subtree provides the conformance information of MIB objects.

5. Relationship to Other MIB Modules

5.1. Relationship to the IF-MIB

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

The ifOperStatus in the ifTable represents whether the mesh function of the AFBR has been triggered. If the softwire mesh capability is negotiated during the BGP OPEN phase, the mesh function is considered to be started, and the ifOperStatus is "up". Otherwise the ifOperStatus is "down".

In the case of an IPv4-over-IPv6 softwire mesh tunnel, ifInUcastPkts counts the number of IPv6 packets which are sent to the virtual interface for decapsulation into IPv4. The ifOutUcastPkts counts the number of IPv6 packets which are generated by encapsulating IPv4 packets sent to the virtual interface. Particularly, if these IPv4 packets need fragmentation, ifOutUcastPkts counts the number of packets after fragmentation.

In the case of an IPv6-over-IPv4 softwire mesh tunnel, ifInUcastPkts counts the number of IPv4 packets, which are delivered up to the virtual interface for decapsulation into IPv6. The ifOutUcastPkts counts the number of IPv4 packets, which are generated by encapsulating IPv6 packets sent down to the virtual interface. Particularly, if these IPv6 packets need to be fragmented, ifOutUcastPkts counts the number of packets after fragmentation. Similar definitions apply to other counter objects in the ifTable.

5.2. Relationship to the IP Tunnel MIB

The IP Tunnel MIB [RFC4087] contains objects applicable to all IP tunnels, including softwire mesh tunnels. Meanwhile, the Softwire Mesh MIB extends the IP Tunnel MIB to further describe encapsulation-specific information.

When running a point to multi-point tunnel, it is necessary for a softwire mesh AFBR to maintain an encapsulation table in order to perform correct "forwarding" among AFBRs. This forwarding function on an AFBR is performed by using the E-IP destination address to look up in the encapsulation table for the I-IP encapsulation destination address. An AFBR also needs to know the BGP peer information of the other AFBRs, so that it can negotiate the NLRI-NH information and the tunnel parameters with them.

The Softwire mesh MIB requires the implementation of the IP Tunnel MIB. The tunnelIfEncapsMethod in the tunnelIfEntry MUST be set to softwireMesh("xx"), and a corresponding entry in the softwire mesh MIB module will be presented for the tunnelIfEntry. The tunnelIfRemoteInetAddress MUST be set to "0.0.0.0" for IPv4 or "::" for IPv6 because it is a point to multi-point tunnel.

-- RFC Ed.: Please replace "xx" with IANA assigned number here.

The tunnelIfAddressType in the tunnelIfTable represents the type of address in the corresponding tunnelIfLocalInetAddress and tunnelIfRemoteInetAddress objects. The tunnelIfAddressType is identical to swmEncapsIIPDstType in softwire mesh, which can support either IPv4-over-IPv6 or IPv6-over-IPv4. When the swmEncapsEIPDstType is IPv6 and the swmEncapsIIPDstType is IPv4, the tunnel type is IPv6-over-IPv4; When the swmEncapsEIPDstType is IPv4 and the swmEncapsIIPDstType is IPv6, the encapsulation mode would be IPv4-over-IPv6.

5.3. MIB modules required for IMPORTS

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

6. Definitions

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

 IMPORTS
     MODULE-IDENTITY, OBJECT-TYPE, mib-2 FROM SNMPv2-SMI
```

```
     OBJECT-GROUP, MODULE-COMPLIANCE            FROM SNMPv2-CONF

     InetAddress, InetAddressType, InetAddressPrefixLength

     FROM INET-ADDRESS-MIB

     ifIndex                                    FROM IF-MIB

     IANAtunnelType                             FROM IANAifType-MIB;

     swmMIB MODULE-IDENTITY
     LAST-UPDATED "201512190000Z"        -- December 19, 2015
     ORGANIZATION "Softwire Working Group"
     CONTACT-INFO "

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

                   Jiang Dong
                   Email:  knight.dongjiang@gmail.com

                   Peng Wu
                   Email:  weapon9@gmail.com

                   Mingwei Xu
                   Email:  xmw@cernet.edu.cn

                   Antti Yla-Jaaski
                   Email:  antti.yla-jaaski@aalto.fi

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

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

                 Copyright (C) The Internet Society (2015).  This
                 version of this MIB module is part of RFC 5565;
                 see the RFC itself for full legal notices."


     REVISION    "201512190000Z"
     DESCRIPTION
                "The MIB module is defined for management of object in
                 the Softwire mesh framework."
     ::= { mib-2 xxx }
```

```
     --RFC Ed.: Please replace "xxx" with IANA assigned number here.

 swmObjects OBJECT IDENTIFIER ::= { swmMIB 1 }

 -- swmSupportedTunnelTable
 swmSupportedTunnelTable OBJECT-TYPE
     SYNTAX      SEQUENCE OF SwmSupportedTunnelEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "A table of objects that shows what kind of tunnels
         can be supported by the AFBR."
     ::= { swmObjects 1 }

 swmSupportedTunnelEntry  OBJECT-TYPE
     SYNTAX      SwmSupportedTunnelEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "A set of objects that show what kind of tunnels
         can be supported in the AFBR. If the AFBR supports
         multiple tunnel types, the swmSupportedTunnelTable
         would have several entries."
     INDEX { swmSupportedTunnelType }
     ::= { swmSupportedTunnelTable 1 }

 SwmSupportedTunnelEntry ::= SEQUENCE {
     swmSupportedTunnelType              IANAtunnelType
 }

 swmSupportedTunnelType OBJECT-TYPE
     SYNTAX      IANAtunnelType
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION
         "Represents the tunnel type that can be used for softwire
         mesh scenarios, such as L2TPv3 over IP, GRE, Transmit
         tunnel endpoint, IPsec in Tunnel-mode, IP in IP tunnel with
         IPsec Transport Mode, MPLS-in-IP tunnel with IPsec Transport
         Mode and IP in IP. There is no restriction of tunnel type
         the Softwire mesh can use."
      REFERENCE
         "L2TPv3 over IP, GRE, IP in IP in RFC5512.
         Transmit tunnel endpoint, IPsec in Tunnel-mode, IP in IP
         tunnel with IPsec Transport Mode, MPLS-in-IP tunnel with
         IPsec Transport Mode in RFC5566."
     ::= { swmSupportedTunnelEntry 1 }
```

```
 -- end of swmSupportedTunnelTable

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

 swmEncapsEntry  OBJECT-TYPE
     SYNTAX      SwmEncapsEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "A table of objects that manage the softwire mesh I-IP
          encapsulation destination based on the E-IP destination
          prefix."
     INDEX { ifIndex,
             swmEncapsEIPDstType,
             swmEncapsEIPDst,
             swmEncapsEIPPrefixLength
           }
     ::= { swmEncapsTable 1 }

 SwmEncapsEntry ::=      SEQUENCE {
     swmEncapsEIPDstType       InetAddressType,
     swmEncapsEIPDst           InetAddress,
     swmEncapsEIPPrefixLength  InetAddressPrefixLength,
     swmEncapsIIPDstType       InetAddressType,
     swmEncapsIIPDst           InetAddress
 }

 swmEncapsEIPDstType OBJECT-TYPE
     SYNTAX      InetAddressType
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "This object specifies the address type used for
          swmEncapsEIPDst. It is different from the tunnelIfAddressType
          in the tunnelIfTable. The swmEncapsEIPDstType is IPv6 (2)
          if it is IPv6-over-IPv4 tunneling. The swmEncapsEIPDstType is
          IPv4 (1) if it is IPv4-over-IPv6 tunneling."
     REFERENCE
         "IPv4 and IPv6 in RFC 4001."
     ::= { swmEncapsEntry 1 }
```

```
 swmEncapsEIPDst OBJECT-TYPE
     SYNTAX      InetAddress
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "The E-IP destination prefix, which is
         used for I-IP encapsulation destination looking up.
         The type of this address is determined by the
         value of swmEncapsEIPDstType"
     REFERENCE
         "E-IP and I-IP in RFC 5565."
     ::= { swmEncapsEntry 2 }

 swmEncapsEIPPrefixLength OBJECT-TYPE
     SYNTAX      InetAddressPrefixLength
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "The prefix length of the E-IP destination prefix."
     ::= { swmEncapsEntry 3 }

 swmEncapsIIPDstType OBJECT-TYPE
     SYNTAX      InetAddressType
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION
         "This object specifies the address type used for
          swmEncapsIIPDst. It is the same as the tunnelIfAddressType
          in the tunnelIfTable."
     REFERENCE
         "IPv4 and IPv6 in RFC 4001."
     ::= { swmEncapsEntry 4 }

 swmEncapsIIPDst OBJECT-TYPE
     SYNTAX      InetAddress
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION
         "The I-IP destination address, which is used as the
         encapsulation destination for the corresponding E-IP
         prefix. Since the tunnelIfRemoteInetAddress in the
         tunnelIfTable should be 0.0.0.0 or ::, swmEncapIIPDst
         should be the destination address used in the outer
         IP header."
     REFERENCE
         "E-IP and I-IP in RFC 5565."
     ::= { swmEncapsEntry 5 }
 -- End of swmEncapsTable
```

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

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

 SwmBGPNeighborEntry ::= SEQUENCE {
         swmBGPNeighborInetAddressType    InetAddressType,
         swmBGPNeighborInetAddress        InetAddress,
         swmBGPNeighborTunnelType         IANAtunnelType
 }

 swmBGPNeighborInetAddressType OBJECT-TYPE
     SYNTAX      InetAddressType
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "This object specifies the address type used for
          swmBGPNeighborInetAddress."
     ::= { swmBGPNeighborEntry 1 }

 swmBGPNeighborInetAddress OBJECT-TYPE
     SYNTAX      InetAddress
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "The address of the AFBR's BGP neighbor. The
         address type is the same as the tunnelIfAddressType
         in the tunnelIfTable."
     ::= { swmBGPNeighborEntry 2 }
```

```
 swmBGPNeighborTunnelType OBJECT-TYPE
     SYNTAX      IANAtunnelType
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION
         "Represents the type of tunnel that the AFBR
         chooses to transmit traffic with another AFBR/BGP
         neighbor."
     ::= { swmBGPNeighborEntry 3 }
 -- End of swmBGPNeighborTable


 -- conformance information
 swmConformance
                     OBJECT IDENTIFIER ::= { swmMIB 2 }
 swmCompliances
                     OBJECT IDENTIFIER ::= { swmConformance 1 }
 swmGroups
                     OBJECT IDENTIFIER ::= { swmConformance 2 }

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

        The following index objects cannot be added as OBJECT
        clauses but nevertheless have compliance requirements:
        "
        -- OBJECT  swmEncapsEIPDstType
        -- SYNTAX  InetAddressType { ipv4(1), ipv6(2) }
        -- DESCRIPTION
        -- "An implementation is required to support
        --  global IPv4 and/or IPv6 addresses, depending
        --  on its support for IPv4 and IPv6."

        -- OBJECT  swmEncapsEIPDst
        -- SYNTAX  InetAddress (SIZE(4|16))
        -- DESCRIPTION
        -- "An implementation is required to support
        --  global IPv4 and/or IPv6 addresses, depending
        --  on its support for IPv4 and IPv6."

        -- OBJECT  swmEncapsEIPPrefixLength
        -- SYNTAX  InetAddressPrefixLength (Unsigned32 (0..128))
        -- DESCRIPTION
        -- "An implementation is required to support
```

```
        --  global IPv4 and/or IPv6 addresses, depending
        --  on its support for IPv4 and IPv6."

        -- OBJECT  swmBGPNeighborInetAddressType
        -- SYNTAX  InetAddressType { ipv4(1), ipv6(2) }
        -- DESCRIPTION
        -- "An implementation is required to support
        --  global IPv4 and/or IPv6 addresses, depending
        --  on its support for IPv4 and IPv6."

        -- OBJECT  swmBGPNeighborInetAddress
        -- SYNTAX  InetAddress (SIZE(4|16))
        -- DESCRIPTION
        -- "An implementation is required to support
        --  global IPv4 and/or IPv6 addresses, depending
        --  on its support for IPv4 and IPv6."

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

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

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

 swmBGPNeighborGroup    OBJECT-GROUP
    OBJECTS {
```

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

 END
```

7. Security Considerations

Because this MIB module reuses the IP tunnel MIB, the security considerations of the IP tunnel MIB is also applicable to the Softwire mesh MIB.

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 objects and their sensitivity/ vulnerability.

Particularly, swmSupportedTunnelType, swmEncapsIIPDstType, swmEncapsIIPDst and swmBGPNeighborTunnelType can expose the types of tunnels used within the internal network, and potentially reveal the topology of the internal network.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), 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.

Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model

(TSM)[RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

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, and the following IANA-assigned tunnelType values recorded in the IANAtunnelType-MIB registry:

```
        Descriptor        OBJECT IDENTIFIER value
        ----------        -----------------------
        swmMIB            { mib-2 xxx }

        IANAtunnelType ::= TEXTUAL-CONVENTION
              SYNTAX     INTEGER {
                         softwireMesh ("xx")  -- softwire Mesh tunnel
                         }
```

9. Acknowledgements

The authors would like to thank Dave Thaler, Jean-Philippe Dionne, Qi Sun, Sheng Jiang, Yu Fu for their valuable comments.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <http://www.rfc-editor.org/info/rfc2578>.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, DOI 10.17487/RFC2579, April 1999, <http://www.rfc-editor.org/info/rfc2579>.

[RFC2580] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance Statements for SMIv2", STD 58, RFC 2580, DOI 10.17487/RFC2580, April 1999, <http://www.rfc-editor.org/info/rfc2580>.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, DOI 10.17487/RFC4001, February 2005, <http://www.rfc-editor.org/info/rfc4001>.

[RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, DOI 10.17487/RFC3414, December 2002, <http://www.rfc-editor.org/info/rfc3414>.

[RFC3826] Blumenthal, U., Maino, F., and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, DOI 10.17487/RFC3826, June 2004, <http://www.rfc-editor.org/info/rfc3826>.

[RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, DOI 10.17487/RFC5512, April 2009, <http://www.rfc-editor.org/info/rfc5512>.

[RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, DOI 10.17487/RFC5565, June 2009, <http://www.rfc-editor.org/info/rfc5565>.

[RFC5566] Berger, L., White, R., and E. Rosen, "BGP IPsec Tunnel Encapsulation Attribute", RFC 5566, DOI 10.17487/RFC5566, June 2009, <http://www.rfc-editor.org/info/rfc5566>.

[RFC5591] Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 5591, DOI 10.17487/RFC5591, June 2009, <http://www.rfc-editor.org/info/rfc5591>.

[RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, DOI 10.17487/RFC5592, June 2009, <http://www.rfc-editor.org/info/rfc5592>.

[RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 6353, DOI 10.17487/RFC6353, July 2011, <http://www.rfc-editor.org/info/rfc6353>.

10.2. Informative References

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <http://www.rfc-editor.org/info/rfc2863>.

[RFC4925] Li, X., Ed., Dawkins, S., Ed., Ward, D., Ed., and A. Durand, Ed., "Softwire Problem Statement", RFC 4925, DOI 10.17487/RFC4925, July 2007, <http://www.rfc-editor.org/info/rfc4925>.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, DOI 10.17487/RFC3410, December 2002, <http://www.rfc-editor.org/info/rfc3410>.

[RFC4087] Thaler, D., "IP Tunnel MIB", RFC 4087, DOI 10.17487/RFC4087, June 2005, <http://www.rfc-editor.org/info/rfc4087>.

Authors' Addresses

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

Phone: +86-10-6260-3059
EMail: yong@csnet1.cs.tsinghua.edu.cn

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

Phone: +86-10-6278-5822
EMail: knight.dongjiang@gmail.com

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

Phone: +86-10-6278-5822
EMail: weapon9@gmail.com

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

Antti Yla-Jaaski
Aalto University
Konemiehentie 2
Espoo 02150
Finland

Phone: +358-40-5954222
EMail: antti.yla-jaaski@aalto.fi

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: October 29, 2013

Sheng Jiang (Editor)
Yu Fu
Bing Liu
Huawei Technologies Co., Ltd
Peter Deacon
IEA Software, Inc.
April 27, 2013

RADIUS Attribute for MAP

draft-jiang-softwire-map-radius-04.txt

Status of this Memo

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

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

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

This Internet-Draft will expire on October 29, 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.

Abstract

Mapping of Address and Port (MAP) is a stateless mechanism for running IPv4 over IPv6-only infrastructure. It provides both IPv4 and IPv6 connectivity services simultaneously during the IPv4/IPv6 co-existing period. The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) MAP options has been defined to configure MAP Customer Edge (CE). However, in many networks, the configuration information may be stored in Authentication Authorization and Accounting (AAA) servers while user configuration is mainly from Broadband Network Gateway (BNG) through DHCPv6 protocol. This document defines a Remote Authentication Dial In User Service (RADIUS) attribute that carries MAP configuration information from AAA server to BNG. The MAP RADIUS attribute are designed following the simplify principle. It provides just enough information to form the correspondent DHCPv6 MAP option.

Table of Contents

1. Introduction 3
2. Terminology 3
3. MAP Configuration process with RADIUS 3
4. Attributes 6
   4.1. MAP-Configuration Attribute 6
   4.2. MAP Rule Options 6
   4.3. Sub Options for MAP Rule Option 7
      4.3.1. Rule-IPv6-Prefix Sub Option 7
      4.3.2. Rule-IPv4-Prefix Sub Option 8
      4.3.3. Encapsulation/Translation Flag Sub Option..... 9
      4.3.4. PSID Sub Option 10
      4.3.5. PSID Length Sub Option 10
      4.3.6. PSID Offset Sub Option 11
   4.4. Table of attributes 11
5. Diameter Considerations 12
6. Security Considerations 12
7. IANA Considerations 12
8. Acknowledgments 12
9. References 13
   9.1. Normative References 13
   9.2. Informative References 13

1. Introduction

Recently providers start to deploy IPv6 and consider how to transit to IPv6. Mapping of Address and Port (MAP) [I-D.ietf-softwire-map] is a stateless mechanism for running IPv4 over IPv6-only infrastructure. It provides both IPv4 and IPv6 connectivity services simultaneously during the IPv4/IPv6 co-existing period. MAP has adopted Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [RFC3315] as auto-configuring protocol. The MAP Customer Edge (CE) uses the DHCPv6 extension options [I-D.mdt-softwire-map-dhcp-option] to discover MAP Border Relay (in tunnel model only) and to configure relevant MAP rules.

In many networks, user configuration information may be managed by AAA (Authentication, Authorization, and Accounting) servers. Current AAA servers communicate using the Remote Authentication Dial In User Service (RADIUS) [RFC2865] protocol. In a fixed line broadband network, the Broadband Network Gateways (BNGs) act as the access gateway of users. The BNGs are assumed to embed a DHCPv6 server function that allows them to locally handle any DHCPv6 requests initiated by hosts.

Since the MAP configuration information is stored in AAA servers and user configuration is mainly through DHCPv6 protocol between BNGs and hosts/CEs, new RADIUS attributes are needed to propagate the information from AAA servers to BNGs. The MAP RADIUS attribute are designed following the simplify principle, while providing enough information to form the correspondent DHCPv6 MAP option. [I-D.mdt-softwire-map-dhcp-option].

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [RFC2119].

The terms MAP CE and MAP Border Relay are defined in [I-D.ietf-softwire-map].

3. MAP Configuration process with RADIUS

The below Figure 1 illustrates how the RADIUS protocol and DHCPv6 cooperate to provide MAP CE with MAP configuration information.

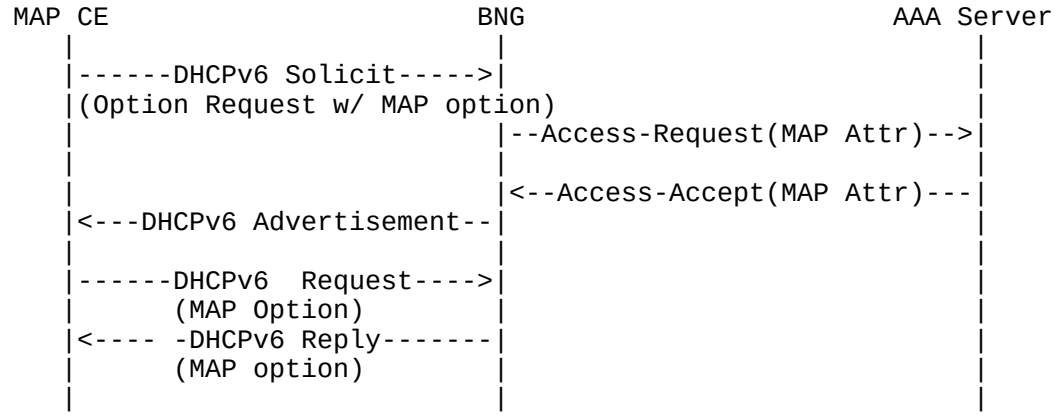


Figure 1: the cooperation between DHCPv6 and RADIUS combining with RADIUS authentication

BNGs act as a RADIUS client and as a DHCPv6 server. First, the MAP CE MAY initiate a DHCPv6 Solicit message that includes an Option Request option (6) [RFC3315] with the MAP option [draft-ietf-softwire-map-dhcp] from the MAP CE. But note that the ORO (Option Request option) with the MAP option could be optional if the network was planned as MAP-enabled as default. When BNG receives the SOLICIT, it SHOULD initiates radius Access-Request message, in which the User-Name attribute (1) SHOULD be filled by the MAP CE MAC address, to the RADIUS server and the User-password attribute (2) SHOULD be filled by the shared MAP password that has been preconfigured on the DHCPv6 server, requesting authentication as defined in [RFC2865] with MAP-Configuration attribute, defined in the next Section. If the authentication request is approved by the AAA server, an Access-Accept message MUST be acknowledged with the IPv6-MAP-Configuration Attribute. After receiving the Access-Accept message with MAP-Configuration Attribute, the BNG SHOULD respond the user an Advertisement message. Then the user can requests for a MAP Option, the BNG SHOULD reply the user with the message containing the MAP option. The recommended format of the MAC address is as defined in Calling-Station-Id (Section 3.20 in [RFC3580]) without the SSID (Service Set Identifier) portion.

Figure 2 describes another scenario, in which the authorization operation is not coupled with authentication. Authorization relevant to MAP is done independently after the authentication process. As similar to above scenario, the ORO with the MAP option in the initial DHCPv6 request could be optional if the network was planned as MAP-enabled as default.

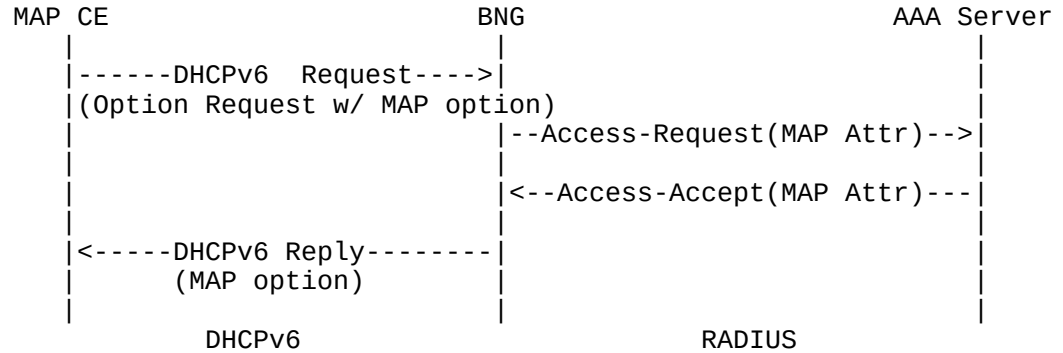


Figure 2: the cooperation between DHCPv6 and RADIUS decoupled with RADIUS authentication

In the abovementioned scenario, the Access-Request packet SHOULD contain a Service-Type attribute (6) with the value Authorize Only (17); thus, according to [RFC5080], the Access-Request packet MUST contain a State attribute that obtained from the previous authentication process.

In both above-mentioned scenarios, Message-authenticator (type 80) [RFC2865] SHOULD be used to protect both Access-Request and Access-Accept messages.

After receiving the MAP-Configuration Attribute in the initial Access-Accept, the BNG SHOULD store the received MAP configuration parameters locally. When the MAP CE sends a DHCPv6 Request message to request an extension of the lifetimes for the assigned address, the BNG does not have to initiate a new Access-Request towards the AAA server to request the MAP configuration parameters. The BNG could retrieve the previously stored MAP configuration parameters and use them in its reply.

If the BNG does not receive the MAP-Configuration Attribute in the Access-Accept it MAY fallback to a pre-configured default MAP configuration, if any. If the BNG does not have any pre-configured default MAP configuration or if the BNG receives an Access-Reject, the tunnel cannot be established.

As specified in [RFC3315], section 18.1.4, "Creation and Transmission of Rebind Messages ", if the DHCPv6 server to which the DHCPv6 Renew message was sent at time T1 has not responded by time T2, the MAP CE (DHCPv6 client) SHOULD enters the Rebind state and attempt to contact any available server. In this situation, the secondary BNG receiving the DHCPv6 message MUST initiate a new Access-Request towards the AAA

server. The secondary BNG MAY include the MAP-Configuration Attribute in its Access-Request.

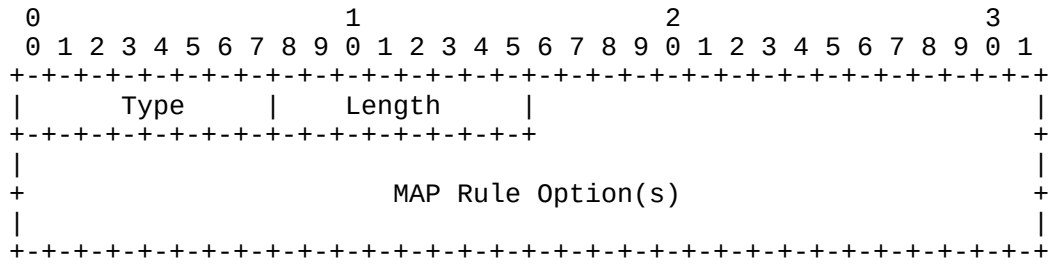
4. Attributes

This section defines MAP-Rule Attribute which is used in the MAP scenario. The attribute design follows [RFC6158] and referring to [I-D.ietf-radext-radius-extensions].

The MAP RADIUS attribute are designed following the simplify principle. The sub options are organized into two categories: the necessary and the optional.

4.1. MAP-Configuration Attribute

The MAP-Configuration Attribute is structured as follows:



Type

TBD

Length

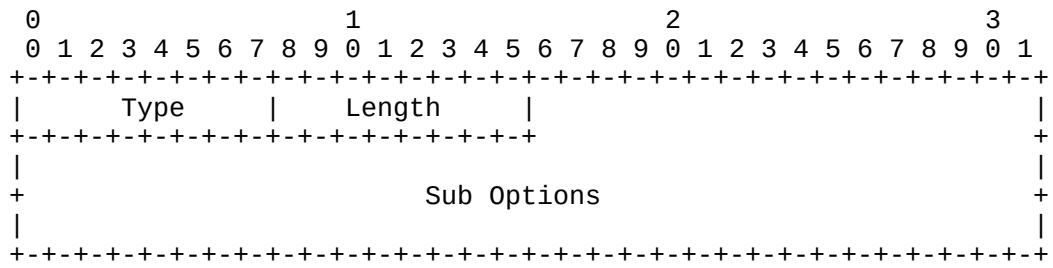
2 + the length of the Rule option(s)

MAP Rule Option (s)

A variable field that may contains one or more Rule option(s), defined in Section 4.2.

4.2. MAP Rule Options

Depending on deployment scenario, one Default Mapping rule and zero or more other type Mapping Rules MUST be included in one MAP-Configuration Attribute.



Type

1 Basic Mapping Rule (Not Forwarding Mapping Rule)

2 Forwarding Mapping Rule (Not Basic Mapping Rule)

3 Default Mapping Rule

4 Basic & Forwarding Mapping Rule

Length

2 + the length of the sub options

Sub Option

A variable field that contains necessary sub options defined in Section 4.3 and zero or several optional sub options, defined in Section 4.4.

4.3. Sub Options for MAP Rule Option

The sub options do not include EA-Len Embedded-Address length , because it can be calculated by the combine of prefix4len, prefix6-len, PSID and offset bits.

4.3.1. Rule-IPv6-Prefix Sub Option

The Rule-IPv6-Prefix Sub Option is necessary for every MAP Rule option. It should appear for once and only once.

The IPv6 Prefix sub option is follow the framed IPv6 prefix designed in [RFC3162].

SubType

1 (SubType number, for the Rule-IPv6-Prefix6 sub option)

SubLen

20 (the length of the Rule-IPv6-Prefix6 sub option)

Reserved

Reserved for future usage. It should be set to all zero.

prefix6-len

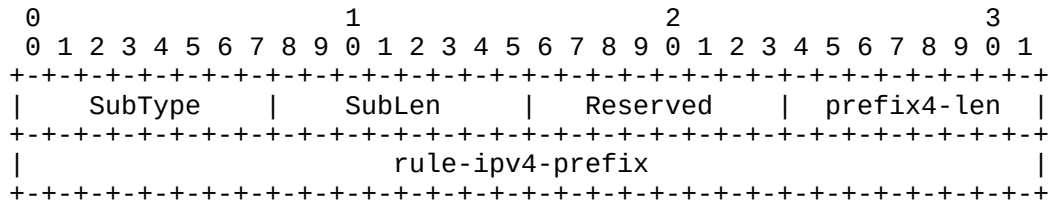
length of the IPv6 prefix, specified in the rule-ipv6-prefix field, expressed in bits

rule-ipv6-prefix

a 128-bits field that specifies an IPv6 prefix that appears in a MAP rule

"For the encapsulation mode the Rule IPv6 prefix can be the full IPv6 address of the BR." [I-D.ietf-softwire-map]

4.3.2. Rule-IPv4-Prefix Sub Option



SubType

2 (SubType number, for the Rule-IPv4-Prefix6 sub option)

SubLen

8 (the length of the Rule-IPv4-Prefix6 sub option)

Reserved

Reserved for future usage. It should be set to all zero.

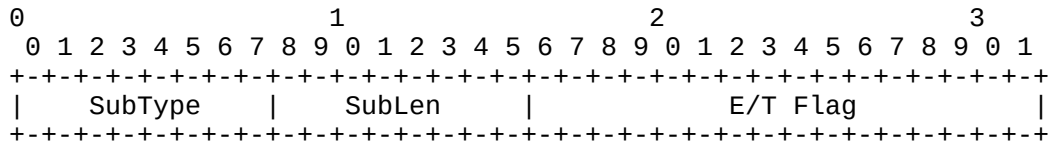
Prefix4-len

length of the IPv6 prefix, specified in the rule-ipv6-prefix field, expressed in bits

rule-ipv4-prefix

a 32-bits field that specifies an IPv4 prefix that appears in a MAP rule

4.3.3. Encapsulation/Translation Flag Sub Option



SubType

3 (SubType number, for the E/T flag sub option)

SubLen

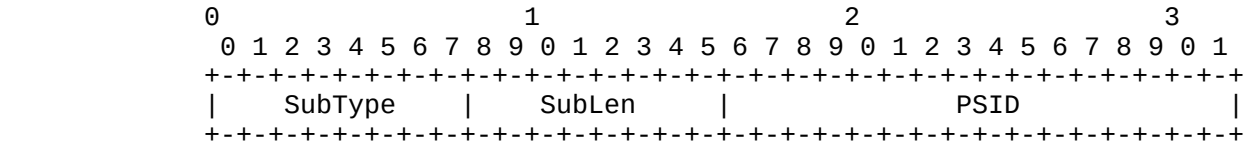
4 (the length of the E/T flag sub option)

E/T Flag

indicate the MAP transport mode: encapsulation or translation. all 0 for encapsulation, all 1 for translation.

If this sub option is not present, the default is to be assumed as encapsulation mode.

4.3.4. PSID Sub Option



SubType

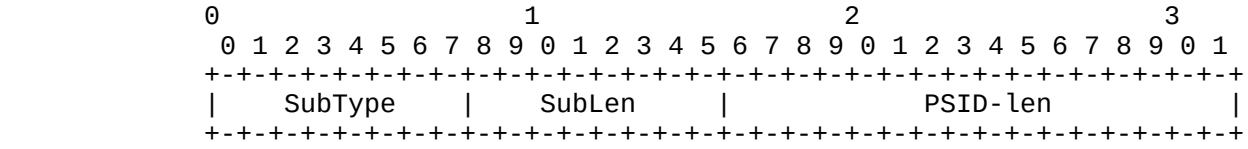
4 (SubType number, for the PSID Sub Option sub option)

SubLen

4 (the length of the PSID Sub Option sub option)

PSID (Port-set ID)
Explicit 16-bit (unsigned word) PSID value. The PSID value algorithmically identifies a set of ports assigned to a CE. The first k-bits on the left of this 2-octets field is the PSID value. The remaining (16-k) bits on the right are padding zeros.

4.3.5. PSID Length Sub Option



SubType

5 (SubType number, for the PSID Length sub option)

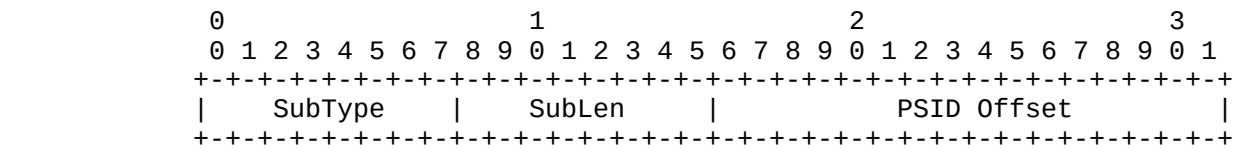
SubLen

4 (the length of the PSID Length sub option)

PSID-len

Bit length value of the number of significant bits in the PSID field. (also known as 'k'). When set to 0, the PSID field is to be ignored. After the first 'a' bits, there are k bits in the port number representing valid of PSID. Subsequently, the address sharing ratio would be 2 ^k.

4.3.6. PSID Offset Sub Option



SubType

6 (SubType number, for the PSID Offset sub option)

SubLen

4 (the length of the PSID Offset sub option)

PSID Offset

4 bits long field that specifies the numeric value for the MAP algorithm's excluded port range/offset bits (A-bits), as per section 5.1.1 in [I-D.ietf-softwire-map]. Default must be set to 4.

4.4. Table of attributes

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

Request	Accept	Reject	Challenge	Accounting Request	#	Attribute
0-1	0-1	0	0	0-1	TBD1	MAP-Configuration
0-1	0-1	0	0	0-1	1	User-Name
0-1	0	0	0	0-1	2	User-Password
0-1	0-1	0	0	0-1	6	Service-Type
0-1	0-1	0-1	0-1	0-1	80	Message-Authenticator

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

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

5. Diameter Considerations

This attribute is usable within either RADIUS or Diameter [RFC6733]. Since the Attributes defined in this document will be allocated from the standard RADIUS type space, no special handling is required by Diameter entities.

6. Security Considerations

In MAP scenarios, both CE and BNG are within a provider network, which can be considered as a closed network and a lower security threat environment. A similar consideration can be applied to the RADIUS message exchange between BNG and the AAA server.

Known security vulnerabilities of the RADIUS protocol are discussed in RFC 2607 [RFC2607], RFC 2865 [RFC2865], and RFC 2869 [RFC2869]. Use of IPsec [RFC4301] for providing security when RADIUS is carried in IPv6 is discussed in RFC 3162 [RFC3162].

A malicious user may use MAC address proofing and/or dictionary attack on the shared MAP password that has been preconfigured on the DHCPv6 server to get unauthorized MAP configuration information.

Security considerations for MAP specific between MAP CE and BNG are discussed in [I-D.ietf-softwire-map]. Furthermore, generic DHCPv6 security mechanisms can be applied DHCPv6 intercommunication between MAP CE and BNG.

Security considerations for the Diameter protocol are discussed in [RFC6733].

7. IANA Considerations

This document requires the assignment of two new RADIUS Attributes Types in the "Radius Types" registry (currently located at http://www.iana.org/assignments/radius-types for the following attributes:

- MAP-Configuration TBD1

IANA should allocate the numbers from the standard RADIUS Attributes space using the "IETF Review" policy [RFC5226].

8. Acknowledgments

The authors would like to thank for valuable comments from Peter Lothberg, Wojciech Dec, and Suresh Krishnan .etc.

9. References

9.1. Normative References

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

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

[RFC3162] Aboba, B., Zorn, G., and D. Mitton, "RADIUS and IPv6", RFC 3162, August 2001.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

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

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

[RFC6158] DeKok, A. and G. Weber, "RADIUS Design Guidelines", RFC 6158, March 2011.

[RFC6733] V. Fajardo, Ed., J. Arkko, J. Loughney, G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, October 2012.

[I-D.ietf-softwire-map] O. Troan, et al., "Mapping of Address and Port (MAP)", draft-ietf-softwire-map, working in progress.

[I-D.mdt-softwire-map-dhcp-option] T. Mrugalski, et al., "DHCPv6 Options for Mapping of Address and Port", draft-mdt-softwire-map-dhcp-option, working in progress.

9.2. Informative References

[RFC2607] Aboba, B. and J. Vollbrecht, "Proxy Chaining and Policy Implementation in Roaming", RFC 2607, June 1999.

[RFC2869] Rigney, C., Willats, W., and P. Calhoun, "RADIUS Extensions", RFC 2869, June 2000.

[I-D.ietf-radext-radius-extensions] DeKok, A. and A. Lior, "Remote Authentication Dial In User Service (RADIUS) Protocol Extensions", draft-ietf-radext-radius-extensions, work in process.

Author's Addresses

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

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

Bing Liu
Huawei Technologies Co., Ltd
Huawei Building, 156 Beiqing Rd.
Hai-Dian District, Beijing 100095
P.R. China
Email: leo.liubing@huawei.com

Peter Deacon
IEA Software, Inc.
P.O. Box 1170
Veradale, WA 99037
USA
EMail: peterd@iea-software.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 10, 2013

Q. Sun
Tsinghua University
Y. Lee
Comcast
Q. Sun
China Telecom
G. Bajko
Nokia
M. Boucadair
France Telecom

October 7, 2012

Dynamic Host Configuration Protocol (DHCP) Option for Port Set Assignment
draft-sun-dhc-port-set-option-00

Abstract

Because of the exhaustion of the IPv4 address space, several techniques have been proposed to share the same IPv4 address among several uses. As an alternative to introducing a level of NAT in the provider's core network, this document provides a mechanism to assign non-overlapping port set to users assigned with the same IPv4 address: Port Set DHCPv4 Option.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 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 3
2. Requirements Language 3
3. DHCPv4 Port Set Option 3
3.1. Port Set Option Format 3
3.2. Port Set Option Example 4
4. Server Behavior 5
5. Client Behavior 5
6. DHCP Unicast Considerations 5
6.1. Server Behavior 6
6.2. Client Behavior 6
7. Security Consideration 6
7.1. Denial-of-Service 6
7.2. Port Randomization 6
8. IANA Consideration 7
9. Contributors List 7
10. References 7
10.1. Normative References 7
10.2. Informative References 8

1. Introduction

Currently some large ISPs still have a large enough IPv4 address pool to be able to allocate public IPv4 addresses for their subscribers. However, due to the exhaustion of the global IPv4 address space, these ISP expect the situation is unsustainable and they will not be able anymore to assign to every requesting host a public IPv4 address.

Two solutions have been proposed so far: (1) Deploy Network Address Translation (NAT) or (2) Allocate the same public IPv4 address with non-overlapped port sets directly to multiple connected devices (which can be CPEs or end hosts). This document focuses on the second solution.

This document describes a new DHCPv4 option which allows the DHCPv4 server to assign a set of ports to a user device during the IPv4 address provisioning process. By assigning the same IPv4 address with non-overlaped port sets to multiple clients, the clients can share the IPv4 address and continue to deliver IPv4 services to subscribers.

The Port Set Option described in this document can be used in various deployment scenarios, some of which are described in [RFC6346]

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. DHCPv4 Port Set Option

3.1. Port Set Option Format

The format of Port Set Option is shown in Figure 1.

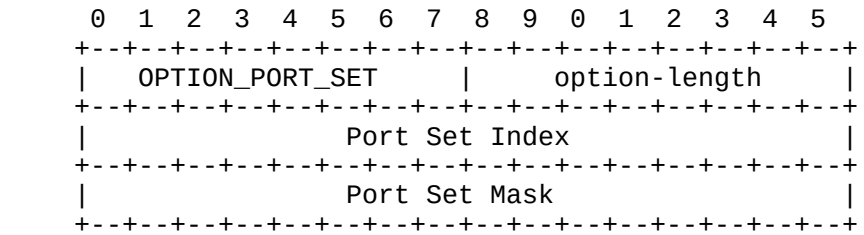


Figure 1 Port Set Option Format

- o option-code: OPTION_PORT_SET (TBD)

- o option-length: An 8-bit field indicating the length of the option excluding the 'Option Code' and the 'Option Length' fields. In this option, the option-length is 4 octets.

- o Port Set Index: Port Set Index identifies a set of ports assigned to a device. The first k bits on the left of the 2-octet field is the Port Set Index value, with the rest of the field right padding zeros.

- o Port Set Mask: Port Set Mask indicates the position of the bits used to build the mask. The first k bits on the left is padding ones while the remained (16-k) bits of the 2-octet field on the right is padding zeros.

In the context of Port Set Option, the port number should consist of port set prefix and port number suffix. The port set prefix can be got from Port Set Index and Port Set Mask, while port number suffix can change continuously. The format of port number is shown in Figure 2.

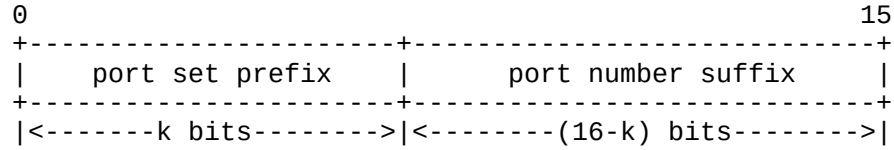


Figure 2 Bit Representation of a port number

In order to exclude the system ports ([I-D.ietf-tsvwg-iana-ports]) or ports saved by SPs, the former port-sets that contains well-known ports SHOULD NOT be assigned.

For example: If k is 10 (the left 10 bits of Port Set Mask is '1'), the first 16 port sets is located in well-known port space, which should not be allocated. Or,

For example: If k is 4 (the left 4 bits of Port Set Mask is '1'), the first port set (0 - 4095) contains the well-know port space. It should be perceived as well.

3.2. Port Set Option Example

The Port Set Option is used to specify one contiguous port set pertaining to the given IP address.

Concretely, this option is used to notify a remote DHCP client about

the port set prefix to be applied when selecting a port value as a source port. The Port Set Option is used to infer a set of allowed contiguous port values. Two port numbers are said to belong to the same Port Set if and only if, they have the same port set prefix.

The following Port Set Index and Port Set Mask are conveyed using DHCP to assign a contiguous port set with excluding well-know ports (with Port Set Index not zero):

Port Set Index: 0001 0100 0000 0000 (5120)

Port Set Mask: 1111 1100 0000 0000 (64512)

The device will get a contiguous port set: 5120 - 6143

4. Server Behavior

The server will not reply with the option until the client has explicitly listed the option code in the Parameter Request List (Option 55).

Server MUST reply with Port Set Option if the client requested OPTION_PORT_SET in its Parameter Request List. The server MUST run an address & port-set pool which plays the same role as address pool in regular DHCP server. The address and port-set pool MUST follow the Port-Mask-format port-set.

The port-set assignment SHOULD be coupled with the address assignment process. Therefore server SHOULD assign the address and port set in the same DHCP messages. And the lease information for the address is applicable to the port-set as well.

5. Client Behavior

The DHCP client applying for the a port-set MUST include either the OPTION_PORT_SET code in the Parameter Request List (Option 55). The client will retrieve a Port Set Option and use the Port Set Index and Port Set Mask to perform the port mask algorithm to get the contiguous port set. The client renews or releases the DHCP lease with the port set.

6. DHCP Unicast Considerations

DHCP messages could be unicasted over UDP port 67. In the context of address sharing, not all the ports are available to the clients. The server cannot use unicast to send the DHCP message to a client which originated the DHCP request. To mitigate this problem, we propose to use the broadcast address (0.0.0.0) when the server replies to the

client. Broadcast address is special and won't be assigned to any client.

6.1. Server Behavior

DHCP server MUST set broadcast bit of the 'flags' field in DHCP messages (Figure 2 of [RFC2131]) when allocating port sets. And DHCP server MUST NOT unicast responses to DHCP client. In order to identify the DHCP responses are sent to which client, client identifier [I-D.ietf-dhc-client-id] is used. DHCP server MUST return client identifier.

6.2. Client Behavior

DHCP client MUST validate client identifier, as specified in [I-D.ietf-dhc-client-id]. DHCP client MUST NOT unicast requests to server: all requests are broadcast. This includes lease renewals. In the case of DHCP relay agent, it will broadcast the server responses to clients.

In some deployment scenarios, DHCP messages containing the proposed DHCP option can be conveyed by other forwarding carrier than IPv4, saying IPv6 [I-D.ietf-dhc-dhcpv4-over-ipv6], etc. The server has to manage to forward DHCP responses to right client.

7. Security Consideration

7.1. Denial-of-Service

The solution is generally vulnerable to DoS when used in shared medium or when access network authentication is not a prerequisite to IP address assignment. The solution SHOULD only be used on point-to-point links, tunnels, and/or in environments where authentication at link layer is performed before IP address assignment, and not shared medium.

7.2. Port Randomization

Preserving port randomization [RFC6056] may be more or less difficult depending on the address sharing ratio (i.e., the size of the port space assigned to a CPE). The host can only randomize the ports inside a fixed port range [RFC6269].

More discussion to improve the robustness of TCP against Blind In-Window Attacks can be found at [RFC5961]. Other means than the (IPv4) source port randomization to provide protection against attacks should be used (e.g., use [I-D.vixie-dnsext-dns0x20] to protect against DNS attacks, [RFC5961] to improve the robustness of

TCP against Blind In-Window Attacks, use IPv6).

A proposal to preserve the entropy when selecting port is discussed in [I-D.bajko-pripaddrassign]

8. IANA Consideration

IANA is kindly requested to allocate DHCP option code to the OPTION_PORT_SET. The code should be added to the DHCP option code space.

9. Contributors List

Many thanks for valuable comments and great efforts from the following contributors:

Peng Wu
Tsinghua University

Email: peng-wu@foxmail.com

Teemu Savolainen
Nokia

Email: teemu.savolainen@nokia.com

Ted Lemon
Nominum, Inc.

Email: mellon@nominum.com

Tina Tsou
Huawei Technologies

Email: tena@huawei.com

Pierre Levis
France Telecom

Email: pierre.levis@orange.com

10. References

10.1. Normative References

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for

Private Internets", BCP 5, RFC 1918, February 1996.

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

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.

[RFC3527] Kinnear, K., Stapp, M., Johnson, R., and J. Kumarasamy, "Link Selection sub-option for the Relay Agent Information Option for DHCPv4", RFC 3527, April 2003.

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

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

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

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

[RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.

10.2. 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.ietf-dhc-client-id] Swamy, N., Halwasia, G., and S. Unit, "Client Identifier Option in DHCP Server Replies", draft-ietf-dhc-client-id-06 (work in progress), October 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-ipv6-05 (work in progress), September 2012.

[I-D.ietf-tsvwg-iana-ports] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", draft-ietf-tsvwg-iana-ports-10 (work in progress), February 2011.

[I-D.vixie-dnsext-dns0x20] Vixie, P. and D. Dagon, "Use of Bit 0x20 in DNS Labels to Improve Transaction Identity", draft-vixie-dnsext-dns0x20-00 (work in progress), March 2008.

Authors' Addresses

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

Phone: +86-10-6278-5822
EMail: sunqi@csnet1.cs.tsinghua.edu.cn

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

Phone:
EMail: yiu_lee@cable.comcast.com

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

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

Gabor Bajko
Nokia

Phone:
EMail: gabor.Bajko@nokia.com

Mohamed Boucadair
France Telecom
2330 Central Expressway
Rennes 35000
France

Phone:
EMail: mohamed.boucadair@orange.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: August 5, 2013

T. Tsou, Ed.
Huawei Technologies (USA)
B. Li
C. Zhou
Huawei Technologies
J. Schoenwaelder
Jacobs University Bremen
R. Penno
Cisco Systems, Inc.
M. Boucadair
France Telecom

February 1, 2013

DS-Lite Failure Detection and Failover
draft-tsou-softwire-bfd-ds-lite-04

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 August 5, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal

Provisions Relating to IETF Documents
(http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Terminology 3
3. Solutions 3
3.1. Bidirectional Forwarding Detection (BFD) 4
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. Internet Control Message Protocol (ICMP) 6
4. Discussion 6
5. Failover 7
6. IANA Considerations 7
7. Security Considerations 7
8. References 8
8.1. Normative References 8
8.2. Informative References 8
Authors' Addresses 9

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 the Internet Control Message Protocol (ICMP). 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.

ICMP: Internet Control Message 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.

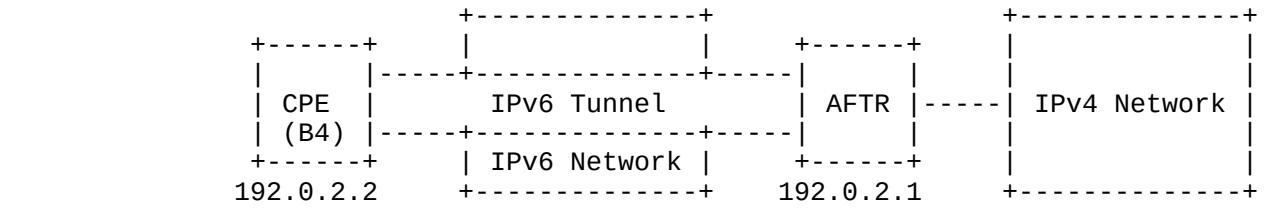


Figure 1: DS-Lite Scenario

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]:

- Peer IP address
- My Discriminator
- Your Discriminator

- Desired Min TX Interval
- Required Min RX Interval
- Required Min Echo RX Interval

In DS-Lite [RFC6334], the B4's WAN-side IPv4 address is the well-known address 192.0.2.2, and the AFTR's well-known IPv4 address is 192.0.2.1, 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

configure the time period between BFD packet transmissions to a longer time, e.g., 10s or 30s.

3.2. Port Control Protocol (PCP)

PCP [I-D.ietf-pcp-base-29] 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.

3.3. Internet Control Message Protocol (ICMP)

The Echo (Request) and Echo Response messages of the Internet Control Message Protocol (ICMP) [RFC0792] [RFC4443] can be used to determine whether remote nodes are reachable. In case of DS-Lite, a B4 element can send Echo (Request) packets to the AFTR periodically. If the B4 element does not receive a certain number (e.g., 3) of Echo Response packets in a certain timeout period, 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.

4. Discussion

The solutions can be compared based on the failure detection time, the overhead on the wire, and the scalability on the AFTR. Lets consider an AFTR that needs to support 10000-30000 subscribers. If every subscriber sends a probe packet every 30 seconds, this creates a load of 1-3 probe packets per millisecond and a failure detection delay in minutes (since multiple probe packets may need to fail in order to detect a failure). Shorter detection times significantly increase the load on AFTRs.

BFD has a simple and fixed packet format, which is easy to implement by logic devices (e.g., ASIC, FPGA). This allows line cards to

process BFD packets very efficiently in hardware.

PCP is a control protocol typically implemented in software. As such, processing a large number of PCP requests in order to detect failures is relatively expensive. On the other hand, PCP can detect the failure of more components of the DS-Lite system. Besides failures of the link and the routing, it also covers certain NAT functions.

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

5. 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.

6. IANA Considerations

This memo includes no request to IANA.

7. 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.

8. References

8.1. Normative References

[I-D.ietf-pcp-base-29]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP) (work in progress)", Nov. 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., Klamm, 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

Cathy Zhou
Huawei Technologies
China

Phone:
Email: cathy.zhou@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 Drivee
San Jose, California 95134
USA

Phone:
Email: repenno@cisco.com

Mohamed Boucadair
France Telecom
Rennes,35000
France

Phone:
Email: mohamed.boucadair@orange.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 16, 2015

X. Xu
Huawei Technologies
N. Sheth
Juniper Networks
R. Asati
Cisco Systems

February 12, 2015

BGP Tunnel Encapsulation Attribute for UDP
draft-xu-softwire-encaps-udp-02

Abstract

This document specifies a new Border Gateway Protocol (BGP) Tunnel Type of User Datagram Protocol (UDP) tunnels.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 16, 2015.

Copyright Notice

Copyright (c) 2015 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 2
1.1. Requirements Language 2
2. Terminology 2
3. BGP Tunnel Type Code for UDP 2
4. Security Considerations 3
5. IANA Considerations 3
6. Contributors 3
7. Acknowledgements 3
8. References 4
8.1. Normative References 4
8.2. Informative References 4
Authors' Addresses 4

1. Introduction

[RFC5512] specifies a method by which Border Gateway Protocol (BGP) speakers can signal tunnel encapsulation information to each other and accordingly it defines support for Generic Routing Encapsulation (GRE) [RFC2784], Layer Two Tunneling Protocol - Version 3 (L2TPv3) [RFC3931] and IP in IP [RFC2003] tunnel types. This document builds on [RFC5512] and defines support for the User Datagram Protocol (UDP) tunnel type which is applicable to the MPLS-in-UDP encapsulation [I-D.ietf-mpls-in-udp].

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Terminology

This memo makes use of the terms defined in [RFC5512].

3. BGP Tunnel Type Code for UDP

To use either the Encapsulation Subsequent Address Family Identifier (SAFI) or the BGP Encapsulation Extended Community defined in [RFC5512] to signal the UDP tunnel type information across BGP speakers, a new Tunnel Type code (TBD) indicating the UDP tunnel type needs to be assigned by IANA. This document does not specify any UDP tunnel specific sub-TLV. Furthermore, the BGP Encapsulation Network Layer Reachability Information (NLRI) Format is not modified by this document.

4. Security Considerations

The security considerations mentioned in [RFC5512] is applicable to this new BGP Tunnel Type code for UDP tunnels as well. No new security risk is introduced by this new Tunnel Type code for UDP tunnels.

5. IANA Considerations

A new BGP Tunnel Type code indicating the UDP tunnel type needs to be assigned by IANA.

6. Contributors

Note that contributors are listed in alphabetical order according to their last names.

Yongbing Fan

China Telecom

Email: fanyb@gsta.com

Yiu Lee

Comcast

Email: Yiu_Lee@Cable.Comcast.com

Zhenbin Li

Huawei Technologies

Email: lizhenbin@huawei.com

7. Acknowledgements

Thanks to

8. References

8.1. Normative References

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

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

8.2. Informative References

[I-D.ietf-mpls-in-udp] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", draft-ietf-mpls-in-udp-11 (work in progress), January 2015.

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.

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

[RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.

Authors' Addresses

Xiaohu Xu
Huawei Technologies
No.156 Beiqing Rd
Beijing 100095
CHINA

Phone: +86-10-60610041
Email: xuxiaohu@huawei.com

Nischal Sheth
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089
USA

Email: nsheth@juniper.net

Rajiv Asati
Cisco Systems
7200 Kit Creek Road
Research Triangle Park,, NC 27709
USA

Email: rajiva@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 6, 2015

X. Xu
Huawei Technologies
R. Asati
Cisco Systems
L. Yong
Huawei USA
Y. Lee
Comcast
Y. Fan
China Telecom
I. Beijnum
Institute IMDEA Networks
December 3, 2014

Encapsulating IP in UDP
draft-xu-softwire-ip-in-udp-03

Abstract

Existing Softwire encapsulation technologies are not adequate for efficient load balancing of Softwire service traffic across IP networks. This document specifies additional Softwire encapsulation technology, referred to as IP-in-User Datagram Protocol (UDP), which can facilitate the load balancing of Softwire service traffic across IP networks.

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 June 6, 2015.

Copyright Notice

Copyright (c) 2014 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 2
1.1. Conventions 3
2. Terminology 3
3. Encapsulation in UDP 3
4. Processing Procedures 5
5. Congestion Considerations 5
6. Security Considerations 6
7. IANA Considerations 7
8. Acknowledgements 7
9. References 8
9.1. Normative References 8
9.2. Informative References 8
Authors' Addresses 9

1. Introduction

To fully utilize the bandwidth available in IP networks and/or facilitate recovery from a link or node failure, load balancing of traffic over Equal Cost Multi-Path (ECMP) and/or Link Aggregation Group (LAG) across IP networks is widely used. [RFC5640] describes a method for improving the load balancing efficiency in a network carrying Softwire Mesh service [RFC5565] over Layer Two Tunneling Protocol - Version 3 (L2TPv3) [RFC3931] and Generic Routing Encapsulation (GRE) [RFC2784] encapsulations. However, this method requires core routers to perform hash calculation on the "load-balancing" field contained in tunnel encapsulation headers (i.e., the Session ID field in L2TPv3 headers or the Key field in GRE headers), which is not widely supported by existing core routers.

Most existing routers in IP networks are already capable of distributing IP traffic "microflows" [RFC2474] over ECMP paths and/or

LAG based on the hash of the five-tuple of User Datagram Protocol (UDP) [RFC0768] and Transmission Control Protocol (TCP) packets (i.e., source IP address, destination IP address, source port, destination port, and protocol). By encapsulating the Softwire service traffic into an UDP tunnel and using the source port of the UDP header as an entropy field, the existing load-balancing capability as mentioned above can be leveraged to provide fine-grained load-balancing of Softwire service traffic traffic over IP networks. This is similar to why LISP [RFC6830] uses UDP encapsulation. Therefore, this specification defines an IP-in-UDP encapsulation method for Software service (including both mesh and hub-spoke modes).

IPv6 flow label has been proposed as an entropy field for load balancing in IPv6 network environment [RFC6438]. However, as stated in [RFC6936], the end-to-end use of flow labels for load balancing is a long-term solution and therefore the use of load balancing using the transport header fields would continue until any widespread deployment is finally achieved. As such, IP-in-UDP encapsulation would still have a practical application value in the IPv6 networks during this transition timeframe.

Similarly, the IP-in-UDP encapsulation format defined in this document by itself cannot ensure the integrity and privacy of data packets being transported through the IP-in-UDP tunnels and cannot enable the tunnel decapsulators to authenticate the tunnel encapsulator. Therefore, in the case where any of the above security issues is concerned, the IP-in-UDP SHOULD be secured with IPsec [RFC4301] or DTLS [RFC6347]. For more details, please see Section 6 of Security Considerations.

1.1. Conventions

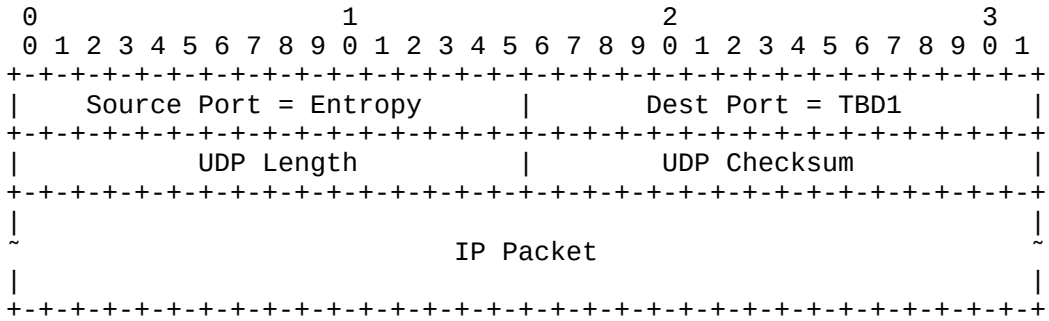
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Terminology

This memo makes use of the terms defined in [RFC5565].

3. Encapsulation in UDP

IP-in-UDP encapsulation format is shown as follows:



Source Port of UDP

This field contains a 16-bit entropy value that is generated by the encapsulator to uniquely identify a flow. What constitutes a flow is locally determined by the encapsulator and therefore is outside the scope of this document. What algorithm is actually used by the encapsulator to generate an entropy value is outside the scope of this document.

In case the tunnel does not need entropy, this field of all packets belonging to a given flow SHOULD be set to a randomly selected constant value so as to avoid packet reordering.

To ensure that the source port number is always in the range 49152 to 65535 (Note that those ports less than 49152 are reserved by IANA to identify specific applications/protocols) which may be required in some cases, instead of calculating a 16-bit hash, the encapsulator SHOULD calculate a 14-bit hash and use those 14 bits as the least significant bits of the source port field while the most significant two bits SHOULD be set to binary 11. That still conveys 14 bits of entropy information which would be enough as well in practice.

Destination Port of UDP

This field is set to a value (TBD1) allocated by IANA to indicate that the UDP tunnel payload is an IP packet. As for whether the encapsulated IP packet is IPv4 or IPv6, it would be determined according to the Version field in the IP header of the encapsulated IP packet.

UDP Length

The usage of this field is in accordance with the current UDP specification [RFC0768].

UDP Checksum

For IPv4 UDP encapsulation, this field is RECOMMENDED to be set to zero because the IPv4 header includes a checksum and use of the UDP checksum is optional with IPv4. For IPv6 UDP encapsulation, the IPv6 header does not include a checksum, so this field MUST contain a UDP checksum that MUST be used as specified in [RFC0768] and [RFC2460] unless one of the exceptions that allows use of UDP zero-checksum mode (as specified in [RFC6935]) applies.

IP Packet

This field contains one IP packet.

4. Processing Procedures

This IP-in-UDP encapsulation causes E-IP[RFC5565] packets to be forwarded across an I-IP [RFC5565] transit core via "UDP tunnels". While performing IP-in-UDP encapsulation, an ingress AFBR (e.g. PE router) would generate an entropy value and encode it in the Source Port field of the UDP header. The Destination Port field is set to a value (TBD1) allocated by IANA to indicate that the UDP tunnel payload is an IP packet. Transit routers, upon receiving these UDP encapsulated IP packets, could balance these packets based on the hash of the five-tuple of UDP packets. Egress AFBRs receiving these UDP encapsulated IP packets MUST decapsulate these packets by removing the UDP header and then forward them accordingly (assuming that the Destination Port was set to the reserved value pertaining to IP).

Similar to all other Softwire tunneling technologies, IP-in-UDP encapsualtion introduces overheads and reduces the effective Maximum Transmision Unit (MTU) size. IP-in-UDP encapsulation may also impact Time-to-Live (TTL) or Hop Count (HC) and Differentiated Services (DSCP). Hence, IP-in-UDP MUST follow the corresponding procedures defined in [RFC2003]. If an ingress AFBR performs fragmentation on an E-IP packet before encapsulating, it MUST use the same source UDP port for all fragmented packets so as to ensures these fragmented packets are always forwarded on the same path.

5. Congestion Considerations

Section 3.1.3 of [RFC5405] discussed the congestion implications of UDP tunnels. As discussed in [RFC5405], because other flows can share the path with one or more UDP tunnels, congestion control [RFC2914] needs to be considered. As specified in [RFC5405]:

> "IP-based traffic is generally assumed to be congestion-controlled, i.e., it is assumed that the transport protocols generating IP-based traffic at the sender already employ mechanisms that are sufficient to address congestion on the path. Consequently, a tunnel carrying IP-based traffic should already interact appropriately with other traffic sharing the path, and specific congestion control mechanisms for the tunnel are not necessary".

Since IP-in-UDP is only used to carry IP traffic which is generally assumed to be congestion controlled, it generally does not need additional congestion control mechanisms.

6. Security Considerations

The security problems faced with the IP-in-UDP tunnel are exactly the same as those faced with IP-in-IP [RFC2003] and IP-in-GRE tunnels [RFC2784]. In other words, the IP-in-UDP tunnel as defined in this document by itself cannot ensure the integrity and privacy of data packets being transported through the IP-in-UDP tunnel and cannot enable the tunnel decapsulator to authenticate the tunnel encapsulator. In the case where any of the above security issues is concerned, the IP-in-UDP tunnel SHOULD be secured with IPsec or DTLS. IPsec was designed as a network security mechanism and therefore it resides at the network layer. As such, if the tunnel is secured with IPsec, the UDP header would not be visible to intermediate routers anymore in either IPsec tunnel or transport mode. As a result, the meaning of adopting the IP-in-UDP tunnel as an alternative to the IP-in-GRE or IP-in-IP tunnel is lost. By comparison, DTLS is better suited for application security and can better preserve network and transport layer protocol information. Specifically, if DTLS is used, the destination port of the UDP header will be filled with a value (TBD2) indicating IP with DTLS and the source port can still be used as an entropy field for load-sharing purposes.

If the tunnel is not secured with IPsec or DTLS, some other method should be used to ensure that packets are decapsulated and forwarded by the tunnel tail only if those packets were encapsulated by the tunnel head. If the tunnel lies entirely within a single administrative domain, address filtering at the boundaries can be used to ensure that no packet with the IP source address of a tunnel endpoint or with the IP destination address of a tunnel endpoint can enter the domain from outside. However, when the tunnel head and the tunnel tail are not in the same administrative domain, this may become difficult, and filtering based on the destination address can even become impossible if the packets must traverse the public Internet. Sometimes only source address filtering (but not destination address filtering) is done at the boundaries of an

administrative domain. If this is the case, the filtering does not provide effective protection at all unless the decapsulator of an IP-in-UDP validates the IP source address of the packet.

7. IANA Considerations

One UDP destination port number indicating IP needs to be allocated by IANA:

Service Name: IP-in-UDP

Transport Protocol(s): UDP

Assignee: IESG <iesg@ietf.org>

Contact: IETF Chair <chair@ietf.org>.

Description: Encapsulate IP packets in UDP tunnels.

Reference: This document.

Port Number: TBD1 -- To be assigned by IANA.

One UDP destination port number indicating IP with DTLS needs to be allocated by IANA:

Service Name: IP-in-UDP-with-DTLS

Transport Protocol(s): UDP

Assignee: IESG <iesg@ietf.org>

Contact: IETF Chair <chair@ietf.org>.

Description: Encapsulate IP packets in UDP tunnels with DTLS.

Reference: This document.

Port Number: TBD2 -- To be assigned by IANA.

8. Acknowledgements

Thanks to Vivek Kumar, Carlos Pignataro and Mark Townsley for their valuable comments on the initial idea of this document. Thanks to Andrew G. Malis for his valuable comments on this document.

9. References

9.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.

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

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

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

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, November 2008.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, April 2013.

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, April 2013.

9.2. Informative References

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.

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

[RFC5640] Filsfils, C., Mohapatra, P., and C. Pignataro, "Load-Balancing for Mesh Softwires", RFC 5640, August 2009.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, November 2011.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

Authors' Addresses

Xiaohu Xu
Huawei Technologies
No.156 Beiqing Rd
Beijing 100095
CHINA

Phone: +86-10-60610041
Email: xuxiaohu@huawei.com

Rajiv Asati
Cisco Systems
7200 Kit Creek Road
Research Triangle Park,, NC 27709
USA

Email: rajiva@cisco.com

Lucy Yong
Huawei USA
5340 Legacy Dr
Plano, TX 75025
USA

Email: Lucy.yong@huawei.com

Yiu Lee
Comcast
One Comcast Center
Philadelphia, PA
USA

Phone: Email: Yiu_Lee@Cable.Comcast.com
Email: cpignata@cisco.com

Yongbing Fan
China Telecom
Guangzhou
CHINA

Email: fanyb@gsta.com

Iljitsch van Beijnum
Institute IMDEA Networks
Avda. del Mar Mediterraneo, 22
Leganes,, Madrid 28918
Spain

Email: iljitsch@muada.com