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

Copyright Notice

Table of Contents

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-publi c-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:

- o If only OPTION_AFTR_NAME is received, then the device MUST operate in stateful mode

- o If both OPTION_AFTR_NAME and OPTION_MAP_BIND are received then the device MUST operate in binding mode

- o If one or more OPTION_MAP_RULE options are received, then the customer side device MUST operate in stateless mode

- o 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.

- o 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:

- o 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:

- o Always return the address of the AFTR for the DS-Lite mode.
- o Always return the address of the lwAFTR for the binding mode.
- o 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: December 31, 2014

Y. Cui
Y. Chen
Tsinghua University
June 29, 2014

Unified IPv6 Transition Framework With Flow-based Forwarding
draft-cui-softwire-unified-v6-framework-00

Abstract

This document describes a software defined networking (SDN) based unified IPv6 transition framework. This framework makes use of the flow-based packet forwarding technology, which can simplify the implementation of both control plane and data plane operations of transition mechanisms. The purpose of this work is to provide an integrated and flexible framework to implement and deploy transition mechanisms, such as MAP-E, Lightweight 4over6, etc.

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].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Currently several IPv6 transition mechanisms have been proposed, such as DS-Lite [RFC6333], MAP-E [I-D.ietf-softwire-map], and Lightweight 4over6 [I-D.ietf-softwire-lw4over6]. These mechanisms have been or being implemented by the industry. However, each of these transition mechanisms have some unique features in control or data plane behavior such as IP provisioning, so the mechanisms require different dedicated implementations on both CPE and BR side and it's complicated and a big cost to upgrade existing devices to support new mechanisms.

This document describes a SDN-based IPv6 transition framework. This framework adopts flow-based packet forwarding technology on transition devices, which can simplify the implementation of both control plane and data plane operations of transition mechanisms. A centralized Controller is added into the network to control the forwarding rules of both CPE and BR, leaving other devices in ISP network between CPE and BR as traditional devices. The forwarding

rules of transition devices can be changed easily by changing network applications on the Controller, thus this framework can implement transition scenarios easily, e.g. IPv4 over IPv6 hub and spoke model that Lightweight 4over6 does. Both CPE and BR can be implemented by standardized SDN switches thus no dedicated devices are needed, so it can reduce the cost of deploying new mechanisms for customer networks.

2. Terminology

This document uses the following terms:

Customer Premises Equipment Switch (CPE Switch): A dual-stack L3 switch that performs the flow-based packet forwarding function. It implements the OpenFlow protocol [OpenFlow] It can be the home gateway or BNG.

Border Relay Switch (BR Switch): A dual-stack L3 switch that performs the flow-based packet forwarding function. It implements the OpenFlow protocol [OpenFlow] It is on the boarder of the ISP network and IPv4/ IPv6 Internet.

Controller: A centralized manager that controls both CPE Switch and BR Switch. The Controller is treated as a logical device that can be a single or multiple physical devices.

Flow-based Packet Forwarding: A function that forwards packets according to the forwarding rules specified by a flow table. A typical forwarding rule includes a match field and a action set. The match field specifies a certain flow, which is a set of packets that share some common features (e.g. same source and destination address). The action set specifies the actions that should act on the certain flow (e.g. forward a packet, encapsulate or decapsulate a packet).

3. Framework Overview

The architecture of the proposed unified IPv6 transition framework is illustrated in Figure 1. The customer LAN connects the IPv6 ISP network through CPE Switch. The ISP Network accesses the Internet through BR Switch. Devices in the ISP Network are not required to be managed by the Controller, thus the operator only needs to upgrade

the CPEs and BRs.

```
                            +---------------+
                     ______|  Controller   |______
                    |      |               |      |
                    |      +---------------+      |
                    |                             |
 ______________     v      ________________       v      ______________
|              | +--------+ |               | +--------+ |              |
| Customer LAN |_|  CPE   |_|    ISP IPv6   |_|   BR   |_|    IPv4      |
|              | | Switch | |    Network    | | Switch | |   Internet   |
|______________| +--------+ |_______________| +--------+ |______________|
```

Figure 1 Architecture Overview

Both CPE Switch and BR Switch perform flow-based forwarding, and are managed by the Controller. Since CPE Switch can work as the gateway of customer network, it also needs to support traditional CPE functions, such as be compatible with [RFC7084].

Controller is responsible for controlling the forwarding behavior of both CPE Switch and BR Switch. Controller can be an individual or a cluster of physical devices. The Controller configures the forwarding rules in the flow tables maintained by the CPE Switch and BR Switch according to the packets passed from Switches. There can also be separate physical controllers at CPE side and BR side, but both controllers needs to share their network state so they can work as a single logical controller. The state sharing method is out of the scope of this document.

Controller communicates with Switch using protocols that is able to carry flow information and flow table configuration. Currently we suggest and choose Openflow [OpenFlow] as the best approach, however some changes are proposed to support IPv6 tunneling and port set based forwarding.

4. Control Plane Behavior

The Controller configures both CPE Switch and BR Switch through flow configuration. The switches are configured or pre-installed with default forwarding policy that forwards all unknown flows to the controller. The controller then installs forwarding rules for the specific flow into the switch. All remaining packets of the same flow are processed based on this rule and will not be forwarded to the Controller. The definition of a flow depends on the forwarding policy. For example, in MAP-E or Lightweight 4over6 scenario, the BR

Switch treats all traffic from the same CPE as a single flow.

Some transition mechanisms require provisioning parameters to CPE. For example, DS-Lite, MAP-E and Lightweight 4over6 use DHCPv6 to provision IPv6 address of AFTR/BR to CPE. Lightweight 4over6 and MAP-E provision IPv4 address and port set to CPE. Since these parameters can be represented by flow forwarding rules, in this framework such DHCPv6 based softwire provisioning is not required. Instead these softwire configuration are embedded into forwarding rules and provisioned to CPE Switch through flow configuration, e.g. the IPv4 address of the CPE can be represented as the value of set-field actions.

The connection between the switches and the controller SHOULD be through IPv6. In order to build the IPv6 based connection, the switches MUST be configured with an IPv6 address and the IPv6 address of the Controller. Since the CPE side requires automatic configuration, DHCPv6 is RECOMMENDED for the CPE Switch to get its IPv6 address or prefix, and the controller address.

5. Data Plane Behavior

When received an incoming packet which is the initial packet of an unknown flow, CPE Switch and BR Switch MUST pass the packet to Controller to ask for the forwarding rule. Controller determines how to proceed the flow, and interpret the process into a set of forwarding rule configurations. Controller then installs these configurations to CPE Switch and/or BR Switch. When receiving the subsequent packets of the flow, CPE Switch and BR Switch will apply the same actions to them, according to the forwarding rules in the flow table.

Both CPE Switch and BR Switch MUST support the IPv6 tunneling encapsulation/decapuslation function required in [RFC6333], [I-D.ietf-softwire-lw4over6] and [I-D.ietf-softwire-map] as actions to the incoming IPv4 packets. IPv4-in-IPv6 tunneling SHOULD be implemented by Switches. When it is not supported, other tunneling format such as GRE tunnel can be used. The encapsulation action is specified with two parameters: source and destination IPv6 address.

5.1. Controller-based NAT44

In MAP-E/Lightweight 4over6 CPE, NAT44 is required. In this framework, NAT44 can be simply supported by flow-based forwarding rules. Since protocol header modification is supported by Switches (i.e. set-field action in OpenFlow), NAT44 can be implemented as modification of source/destination IPv4 addresses and port fields. As an example, for the upstreaming traffic on CPE Switch, the Switch

asks for the forwarding rule of each flow identified by (source IPv4 address, source port), and the Controller allocates an address and a port for each flow and orders the Switch to rewrite the source IPv4 address and port of the upstreaming flow, and the destination IPv4 address and port of the corresponding downstreaming flow. The NAT44 state is maintained by the Controller.

5.2. Port-Set Based Packet Matching

Since MAP-E/Lightweight 4over6 requires port-based IPv4 address sharing, there may be thousands of ports of the same IPv4 address allocated to the same CPE. To reduce the amount of forwarding rules in BR Switch, when BR Switch matching an incoming packet, it SHOULD support the port-set based matching as defined in in Section 5.1 of [I-D.ietf-softwire-map]. The BR Switch SHOULD support and accept match field masking for ports (see section 7.2.2.5 of [OpenFlow]). For the PSID usage, the PSID bits (k bits) in the port mask are filled with "1"s, and the remaining bits (a+m bits) are filled with "0"s.

6. Example: 4over6 Mode

This section describes an example of running IPv4 over IPv6 tunneling mode in this framework. The network behavior is similar to Lightweight 4over6, however the DHCPv6-based IPv4 address and PSID provisioning for lwB4 is not needed. In this mode, The Customer LAN is an IPv4 or dual-stack network, and the ISP Network is an IPv6 network. Initially the CPE Switch is configured with IPv6 prefix through DHCPv6 Prefix Delegation and connects to the Controller through OpenFLow. Then the CPE Switch is ordered to pass all unknown packets to the Controller.

The Controller manages the binding between IPv4 addresses and IPv6 addresses. When a new CPE Switch connects to the Controller, the Controller allocates an IPv4 address and a port set for the CPE Switch, and installs per-subscriber scale forwarding rule(s) in BR Switch. To work as MAP-E mode, the IPv4 address and port set are calculated from the CPE’s IPv6 prefix. To work as Lightweight 4over6 mode, the IPv4 address and port set are allocated dynamically.

In case that CPE’s NAT44 state is maintained in Controller, when received an unknown packet from CPE Switch, the Controller picks one port from the port set of the CPE Switch and installs the forwarding rule in the CPE Switch, to rewrite the source IPv4 address and port into public ones on the flow. Both forwarding rules in CPE Switch and BR Switch include IPv4-in-IPv6 tunnel encapsulation/decapuslation action, taking the IPv6 address of the opposite as an parameter.

6.1. Forwarding Rules

A CPE Switch is configured with following rules:

Decapsulation rule: For all IPv6 traffic from ISP network with the protocol type 4, pop the IPv6 tunnel header.

NAT rule: For each flow received from LAN network and identified by (source IPv4 address, source port): rewrite source IPv4 address and source port to public values specified by Controller; also rewrite destination IPv4 address and port of the corresponding opposite flow.

Encapsulation rule: For all IPv4 traffic from LAN network: push a IPv6 tunnel header (protocol type 4), of which the source and destination address are specified by the Controller. The tunnel source address is one of the CPE Switch’s addresses chosen by the Controller. The tunnel destination address is the BR Switch’s address.

A BR Switch is configured with following rules:

Decapsulation rule: For all IPv6 traffic from ISP network with the protocol type 4, pop the IPv6 tunnel header and forward to Internet.

Encapsulation rule: For each flow received from Internet side and identified by (destination IPv4 address, destination port set): push a IPv6 tunnel header (protocol type 4), of which the source and destination address are specified by the Controller. The tunnel source address is the BR Switch’s address. The tunnel destination address is one of the CPE Switch’s address, which MUST be the same address used in CPE Switch’s forwarding rule. By masking a port set mask when doing port matching, one matching rule can match a set of ports.

6.2. Mesh Mode

This framework can support mesh mode for Lightweight 4over6. In mesh mode, if the (destination IPv4 address, destination port) of a LAN side incoming flow on a CPE Switch (CPE1) belongs to another CPE Switch (CPE2), the tunnel destination address of the encapsulation action for this flow is set to CPE2’s IPv6 address directly. The priority of mesh mode encapsulation rule is higher than default encapsulation rule.

7. NAT Consideration

Section 5.1 describes how Controller-based NAT44 works. In some cases that only NAT44 but no other applications are performed for TCP/UDP flows, passing the initial packet of each flow to Controller may be inefficient. In this case, the framework proposed in this document allows the switches to have a dedicated NAT module that handles NAT44 states and rewrites packets. The NAT module works as a virtual interface to the switch, and the switch perform NAT44 action to packets by forwarding packets to the NAT interface. The NAT44 module needs to be configured with the public IPv4 address and/or port set. In case a switch has a dedicated NAT44 module, the Controller may still want to handle the forwarding rules of part of the flows, to provide better quality for some services.

8. Security Considerations

As Switch should always send unknown flows to Controller, the link between Switch and Controller might be vulnerable to a typical DoS attack, which is done by flooding new sessions and can exhaust all available resources of Switch and/or Controller. Some monitoring or filtering approaches should be used to prevent against such an attack. In addition, Controller can implement a policy that restricts the resource allocated to a Switch, or restricts the resource of Switch allocated to a flow.

Further security consideration is TBD.

9. IANA Considerations

This document does not include an IANA request.

10. References

10.1. Normative References

[I-D.ietf-softwire-lw4over6]
Cui, Y., Qiong, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture", draft-ietf-softwire-lw4over6-10 (work in progress), June 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-10 (work in progress), January 2014.

[OpenFlow]
Open Networking Foundation, "OpenFlow Switch Specification 1.4.0", <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf>.

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

10.2. Informative References

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

[RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, November 2013.

Authors' Addresses

Yong Cui
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6260-3059
Email: yong@csnet1.cs.tsinghua.edu.cn

Yuchi Chen
Tsinghua University
Beijing 100084
P.R.China

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

Softwire WG
Internet-Draft
Updates: RFC7568 (if approved)
Intended status: Standards Track
Expires: August 5, 2017

I. Farrer
Deutsche Telekom AG
Q. Sun
Y. Cui
L. Sun
Tsinghua University
February 1, 2017

DHCPv4 over DHCPv6 Source Address Option
draft-fsc-softwire-dhcp4o6-saddr-opt-07

Abstract

DHCPv4 over DHCPv6 [RFC7341] describes a mechanism for dynamically configuring IPv4 over an IPv6-only network. For DHCPv4 over DHCPv6 to function with some IPv4-over-IPv6 softwire mechanisms and deployment scenarios, the operator must learn the /128 IPv6 address that the client will use as the source of IPv4-in-IPv6 tunnel. This address, in conjunction with the IPv4 address and the Port Set ID allocated to the DHCP 4o6 client are used to create a binding table entry in the softwire tunnel concentrator. This memo defines two DHCPv6 options used to communicate the source tunnel IPv6 address between the DHCP 4o6 client and server. It also describes a method for configuring the client with the IPv6 address of the border router so that the softwire can be established. It is designed to work in conjunction with the IPv4 address allocation process.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Deterministic IPv4-over-IPv6 transition technologies require that elements are pre-configured with binding rules for routing traffic to clients. This places a constraint on the location of the client's tunnel endpoint: The tunnel endpoint has to be a pre-determined prefix which is usually be configured on the home gateway device. [RFC7597] describes a DHCPv6 based mechanism for provisioning such deterministic softwires.

A dynamic provisioning model, such as using DHCPv4 over DHCPv6 [RFC7341] allows much more flexibility in the location of the IPv4-over-IPv6 tunnel endpoint, as the IPv6 address is dynamically

signalled back to the service provider so that the corresponding tunnel configuration in the border router (BR) can be created. The DHCP 4o6 client and tunnel client could be run on end devices attached to any routable IPv6 prefix allocated to an end-user, located anywhere within an arbitrary home network topology. Dynamic allocation also helps to optimize IPv4 resource usage as only clients which are currently active are allocated IPv4 addresses.

This document describes a mechanism for dynamically provisioning softwires created using DHCPv4 over DHCPv6 (DHCP 4o6), including provisioning the client with the address of the softwire border router (BR) and informing the service provider of client’s binding between the dynamically allocated IPv4 address and Port Set ID and the IPv6 address that the softwire Initiator will use for accessing IPv4-over-IPv6 services.

It is used with DHCP 4o6 message flows to communicate the binding over the IPv6-only network. The service provider can then use this binding information to provision other functional elements in their network accordingly, e.g. using the client’s binding information to synchronise the binding table in the border router.

2. Applicability

The mechanism described in this document is only suitable for use for provisioning softwire clients via DHCP 4o6. The options described here are only applicable within the DHCP 4o6 message exchange process. Current softwire technologies suitable for extending to incorporate DHCPv4 over DHCPv6 with dynamic IPv4 address leasing include [RFC7597] and [RFC7596].

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 RFC 2119 [RFC2119].

4. Solution Overview

The solution in this document is intended for the dynamic establishment of IPv4-over-IPv6 softwires. DHCP 4o6 [RFC7341] supports dynamically allocating (shared) IPv4 address. For a softwire to be successfully created, the IPv4 address has to be linked to the client’s IPv6 tunnel source address. Within this process, the DHCP 4o6 client uses a DHCPv6 option to signal its tunnel source IPv6 address to the DHCP 4o6 server so that the operator’s provisioning system can create the binding and configure the tunnel concentrator accordingly.

Two new DHCPv6 options are defined in this memo: OPTION_DHCP4O6_SADDR_HINT and OPTION_DHCP4O6_SADDR. They are intended to be used alongside the normal DHCPv4 IPv4 address allocation message flow in the context of DHCP 4o6. If a DHCP 4o6 client supports this mechanism, it MUST include the code of OPTION_DHCP4O6_SADDR_HINT in the Option Request Option (ORO) [RFC3315] when requesting IPv4 configuration through DHCP 4o6.

The communication of parameters between the client and server is a two-way process: OPTION_DHCP4O6_SADDR_HINT is optionally used by the DHCP 4o6 server to indicate to the client a preferred IPv6 prefix for binding the received IPv4 configuration and sourcing tunnel traffic. This may be necessary if there are multiple IPv6 prefixes in use in the customer network (e.g. ULAs), or if the specific IPv4-over-IPv6 transition mechanism requires the use of a particular prefix for any reason. When the client has selected an IPv6 address to bind the IPv4 configuration to, it passes the address back to the DHCP 4o6 server using OPTION_DHCP4O6_SADDR.

4.1. Provisioning the BR Address

To configure a softwire, the initiator also requires the IPv6 address of the BR. Section 4.2 of [RFC7598] defines option 90 (OPTION_S46_BR) for this purpose, but mandates that the option can only be used when encapsulated within one of the softwire container options: OPTION_S46_CONT_MAPE, OPTION_S46_CONT_MAPT or OPTION_S46_CONT_LW. From Section 3 of [RFC7598]:

> "Softwire46 DHCPv6 clients that receive provisioning options that are not encapsulated in container options MUST silently ignore these options."

This document updates [RFC7598] to remove this restriction for DHCPv6 option 90 (OPTION_S46_BR) allowing it to appear directly within the list of options in the client’s ORO request and directly within subsequent messages sent by the DHCPv6 server.

5. IPv6/IPv4 Binding Message Flow

The following diagram shows the client/server message flow and how the options defined in this document are used. In each step, the relevant DHCPv4 message is given above the arrow and the relevant options below the arrow. All the DHCPv4 messages here are encapsulated in DHCPv4-query or DHCPv4-response messages, and those options are included in the ’options’ field of the DHCPv4-query or DHCPv4-response message.

```
        DHCP 4o6                                                DHCP 4o6
         Client                                                  Server
           |                DHCPDISCOVER (DHCPv4)                   |
    Step 1 |------------------------------------------------------>|
           |               ORO with OPTION_S46_BR,                  |
           |          OPTION_DHCP4O6_SADDR_HINT(DHCPv6)             |
           |                                                        |
           |                 DHCPOFFER (DHCPv4)                     |
    Step 2 |<------------------------------------------------------|
           |     OPTION_S46_BR, OPTION_DHCP4O6_SADDR_HINT           |
           |    (cipv6-prefix-hint with service provider's          |
           |              preferred prefix) (DHCPv6)                |
           |                                                        |
           |                 DHCPREQUEST (DHCPv4)                   |
    Step 3 |------------------------------------------------------>|
           |                  OPTION_S46_BR,                        |
           |    OPTION_DHCP4O6_SADDR (cipv6-bound-prefix with       |
           |     client's bound /128 IPv6 address) (DHCPv6)         |
           |                                                        |
           |                  DHCPACK (DHCPv4)                      |
    Step 4 |<------------------------------------------------------|
           |                   OPTION_S46_BR,                       |
           |    OPTION_DHCP4O6_SADDR (cipv6-bound-prefix with       |
           |      client's bound /128 IPv6 prefix) (DHCPv6)         |
           |                                                        |
```

IPv6/IPv4 Binding Message Flow

A client attempting dynamic softwire configuration includes the option code for OPTION_BR_PREFIX, OPTION_DHCP4O6_SADDR_HINT in the DHCPv6 ORO in all DHCPv4-query messages it sends.

When a DHCP 4o6 Server replies with a DHCPOFFER message, it SHOULD include OPTION_S46_BR. It MAY also include OPTION_DHCP4O6_SADDR_HINT, which is used to indicate a preferred prefix that the client should use to bind IPv4 configuration to. If this option is received, the client MUST perform a longest prefix match between cipv6-prefix-hint and all prefixes/addresses in use on the device. If a match is found, the selected prefix MUST then be used to bind the received IPv4 configuration to and source the tunnel from. If no match is found, or the client doesn't receive OPTION_DHCP4O6_SADDR_HINT the client MAY select any valid IPv6 address to use as the tunnel source.

Once the client has selected which prefix it will use, it MAY use either an existing IPv6 address that is already configured on an interface, or create a new address specifically for use as the

softwire source address (e.g. using an Interface Identifier constructed as per Section 6 of [RFC7597]). If a new address is being created, the client MUST complete configuration of the new address, performing duplicate address detection (if required) before proceeding to Step 3.

OPTION_DHCP4O6_SADDR is used by the client to inform the DHCP 4o6 Server which IPv6 address the IPv4 configuration has been bound to. The client MUST put the selected IPv6 softwire source address into this option and include it in the DHCPv4-response message when it sends the DHCPREQUEST message.

6. DHCPv6 Options

6.1. DHCPv4 over DHCPv6 Source Address Hint Option

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   OPTION_DHCP4O6_SADDR_HINT   |         option-length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|cipv6-hintlen  |                                               |
+-+-+-+-+-+-+-+-+          cipv6-prefix-hint                    .
.                          (variable length)                    .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- o option-code: OPTION_DHCP4O6_SADDR_HINT (TBA1)
- o option-length: 1 + length of cipv6-prefix-hint, specified in bytes.
- o cipv6-hintlen: 8-bit field expressing the bit mask length of the IPv6 prefix specified in cipv6-prefix-hint. Valid values are 0 to 128.
- o cipv6-prefix-hint: The IPv6 prefix indicating the preferred prefix for the client to bind the received IPv4 configuration to. The length is (cipv6-hintlen + 7) / 8. The field is padded on the right with zero bits up to the nearest octet boundary when cipv6-prefix-hint is not evenly divisible by 8.

OPTION_DHCP4O6_SADDR_HINT is a singleton. Servers MUST NOT send more than one instance of the OPTION_DHCP4O6_SADDR_HINT option.

6.1.1. Client Option Validation Behavior

On receipt of the OPTION_DHCP4O6_SADDR_HINT option, the client makes the following validation checks:

- o The received cipv6-hintlen value is not larger than 128.

- o The received cipv6-hintlen value is not larger than the number of bytes sent in the cipv6-prefix-hint field. (e.g. the cipv6-hintlen is 128 but the cipv6-prefix-hint has only 8 bytes).

For either of these cases the receiver MAY either discard the option and proceed to attempt configuration as if the option had not been received, or attempt to use the received values for the long prefix match anyway.

The receiver MUST only use bits the cipv6-prefix-hint field up to the value specified in the cipv6-hintlen when performing the longest prefix match. cipv6-prefix-hint bits beyond this value MUST be ignored.

6.2. DHCPv4 over DHCPv6 Source Address Option

The format of DHCPv4 over DHCPv6 Source address option is defined as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     OPTION_DHCP4O6_SADDR      |         option-length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                        cipv6-src-address                      +
.                           (128 bits)                          .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- o option-code: OPTION_DHCP4O6_SADDR (TBA2)
- o option-length: 16.
- o cipv6-src-address: 16 bytes long; The IPv6 address that the client has bound the allocated IPv4 configuration to.

7. Security Considerations

Security considerations which are applicable to [RFC7341] are also applicable here.

8. IANA Considerations

IANA is requested to allocate a DHCPv6 option code for OPTION_DHCP4O6_SADDR_HINT and a DHCPv4 option code for OPTION_DHCP4O6_SADDR.

9. Acknowledgements

The authors would like to thank Ted Lemon, Lishan Li and Tatuya Jinmei for their contributions and 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>.

[RFC7341] Sun, Q., Cui, Y., Siodelski, M., Krishnan, S., and I. Farrer, "DHCPv4-over-DHCPv6 (DHCP 4o6) Transport", RFC 7341, DOI 10.17487/RFC7341, August 2014, <http://www.rfc-editor.org/info/rfc7341>.

10.2. Informative References

[I-D.farrer-softwire-br-multiendpoints] Farrer, I. and Q. Sun, "Multiple BR Tunnel Endpoint Addresses", draft-farrer-softwire-br-multiendpoints-01 (work in progress), July 2015.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <http://www.rfc-editor.org/info/rfc2131>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <http://www.rfc-editor.org/info/rfc3315>.

[RFC7596] Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture", RFC 7596, DOI 10.17487/RFC7596, July 2015, <http://www.rfc-editor.org/info/rfc7596>.

[RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <http://www.rfc-editor.org/info/rfc7597>.

[RFC7598] 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", RFC 7598, DOI 10.17487/RFC7598, July 2015, <http://www.rfc-editor.org/info/rfc7598>.

Authors' Addresses

Ian Farrer
Deutsche Telekom AG
CTO-ATI, Landgrabenweg 151
Bonn, NRW 53227
Germany

Email: ian.farrer@telekom.de

Qi Sun
Tsinghua University
Beijing 100084
P.R. China

Phone: +86-10-6278-5822
Email: sunqi.csnet.thu@gmail.com

Yong Cui
Tsinghua University
Beijing 100084
P.R. China

Phone: +86-10-6260-3059
Email: yong@csnet1.cs.tsinghua.edu.cn

Linhui Sun
Tsinghua University
Beijing 100084
P.R. China

Phone: +86-10-6278-5822
Email: lh.sunlinh@gmail.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2015

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

July 2, 2014

DS-Lite Management Information Base (MIB)
draft-ietf-softwire-dslite-mib-06

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 Dual-Stack Lite (DS-Lite).

Status of This Memo

Copyright Notice

Table of Contents

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 devices 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 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 originated. 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 NAT 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 defined other tunnel types in the future, this DS-Lite MIB will be updated then.

The NAT MIB [I-D.ietf-behave-nat-mib] 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 supports IP in IP tunnels.

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

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

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

In a DS-Lite scenario, the AFTR is not only the tunnel end concentrator, but also a 4-4 translator. Within the Address Family Transition Router (AFTR), tunnel information and translation information MUST be mapped each other. But the tunnel entry defined in the IP Tunnel MIB and the NAT mapping entry defined in the NAT MIB are not able to reflect this mapping relationship. Therefore, a combined MIB is necessary.

The implementation of the IP Tunnel MIB is required for DS-Lite. 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 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 NAT MIB [I-D.ietf-behave-nat-mib].

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 some objects defined in the IP Tunnel MIB are "not access", 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 AFTR which is capable of a NAT function. Because the NAT MIB supports the NAT management function in DS-Lite, we may reuse it in DS-Lite MIB. The dsliteNAT subtree also provides the information of mapping relationship between the tunnel entry and NAT entry by extending the IPv6 address of B4 to the natMappingTableEntry in the NAT 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 DS-Lite.

6.2.1. The dsliteTrap Subtree

The dsliteTrap subtree provides trap information in 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 [RFC4008].

8. Definitions

```
DSLite-MIB DEFINITIONS  ::=  BEGIN

   IMPORTS
```

```
         MODULE-IDENTITY, OBJECT-TYPE, transmission,
         NOTIFICATION-TYPE,Gauge32,TimeTicks,
         Integer32, Counter64,Unsigned32
            FROM SNMPv2-SMI

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

         DisplayString
            FROM SNMPv2-TC

         SnmpAdminString
            FROM SNMP-FRAMEWORK-MIB

         ifIndex
            FROM IF-MIB

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

        ProtocolNumber, NatBehaviorType,
        NatPoolingType, SubscriberIndex
             FROM NAT-MIB;

      dsliteMIB MODULE-IDENTITY
      LAST-UPDATED "201407020000Z"          -- July 02, 2014
      ORGANIZATION "IETF Softwire Working Group"
      CONTACT-INFO
         "Yu Fu
          Huawei Technologies Co., Ltd
          Huawei Building, 156 Beiqing Rd., Hai-Dian District
          Beijing, P.R. China 100095
          EMail: eleven.fuyu@huawei.com

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

          Jiang Dong
          Tsinghua University
          Department of Computer Science, Tsinghua University
          Beijing  100084
          P.R. China
          Email:  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 object in the
         DS-Lite scenario.
         Copyright (C) The Internet Society (2014).  This version
         of this MIB module is part of RFC yyyy; see the RFC itself
         for full legal notices. "
      REVISION    "201407020000Z"
      DESCRIPTION
         "Initial version. Published as RFC xxxx."
   --RFC Ed.: RFC-edtitor pls fill in xxxx
         ::=  {  transmission 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 }

      dsliteTraps  OBJECT IDENTIFIER
            ::=  { dsliteNotifications 1  }

   --dsliteTunnel

   --dsliteTunnelTable

      dsliteTunnelTable OBJECT-TYPE
```

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

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

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

       dsliteTunnelStartAddressType 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 startpoint and endpoint."
          ::= { dsliteTunnelEntry 1 }

      dsliteTunnelStartAddress OBJECT-TYPE
          SYNTAX     InetAddress
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
             "The address of the start point of the tunnel."
          ::= { dsliteTunnelEntry 2 }
```

```
      dsliteTunnelEndAddress OBJECT-TYPE
          SYNTAX     InetAddress
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
             "The address of the endpoint of the tunnel."
          ::= { dsliteTunnelEntry 3 }

      dsliteTunnelStartAddPreLen OBJECT-TYPE
          SYNTAX InetAddressPrefixLength
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
             "IPv6 prefix length of the IP address of the
              start point of the tunnel."
          ::= { dsliteTunnelEntry 4 }



    --dsliteNAT
    --dsliteNATMapTable(define address pool, natPoolTable and
    --natPoolRangeTable defined in draft-ietf-behave-nat-mib
    --are sufficient)
    --dsliteNATBindTable(NAPT)

       dsliteNATBindTable OBJECT-TYPE
         SYNTAX     SEQUENCE OF DsliteNATBindEntry
         MAX-ACCESS not-accessible
         STATUS     current
         DESCRIPTION
            "This table contains information about currently
             active NAT binds in AFTR. This table extends the
             IPv6 address of B4 to the natMappingTable
             designed in NAT MIB(draft-ietf-behave-nat-mib)."
         ::=  { dsliteNAT 1 }

      dsliteNATBindEntry OBJECT-TYPE
         SYNTAX     DsliteNATBindEntry
         MAX-ACCESS not-accessible
         STATUS     current
         DESCRIPTION
            "Each entry in this table holds the relationship between
             tunnel information and nat bind information. These entries
             are lost upon agent restart."
         INDEX   { dsliteNATBindMappingProto,
                   dsliteNATBindMappingExtRealm,
                   dsliteNATBindMappingExtAddressType,
                   dsliteNATBindMappingExtAddress,
```

```
                    dsliteNATBindMappingExtPort,
                    dsliteTunnelStartAddress,
                    dsliteTunnelStartAddPreLen }
         ::=  {  dsliteNATBindTable 1   }

     DsliteNATBindEntry  ::=
         SEQUENCE {
         dsliteNATBindMappingProto          ProtocolNumber,
         dsliteNATBindMappingExtRealm       SnmpAdminString,
         dsliteNATBindMappingExtAddressType InetAddressType,
         dsliteNATBindMappingExtAddress     InetAddress,
         dsliteNATBindMappingExtPort        InetPortNumber,
         dsliteNATBindMappingIntRealm       SnmpAdminString,
         dsliteNATBindMappingIntAddressType InetAddressType,
         dsliteNATBindMappingIntAddress     InetAddress,
         dsliteNATBindMappingIntPort        InetPortNumber,
         dsliteNATBindMappingPool           Unsigned32,
         dsliteNATBindMappingMapBehavior    NatBehaviorType,
         dsliteNATBindMappingFilterBehavior NatBehaviorType,
         dsliteNATBindMappingAddressPooling NatPoolingType,
         dsliteNATBindMappingSubsIndex      SubscriberIndex
         }

      dsliteNATBindMappingProto OBJECT-TYPE
          SYNTAX      ProtocolNumber
          MAX-ACCESS  not-accessible
          STATUS      current
          DESCRIPTION
          " This object specifies the mapping’s transport protocol
          number."
          ::= { dsliteNATBindEntry 1 }

     dsliteNATBindMappingExtRealm OBJECT-TYPE
          SYNTAX     SnmpAdminString (SIZE(0..32))
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
          " The realm to which dsliteNATBindMappingExtAddress belongs."
          ::= { dsliteNATBindEntry 2 }

      dsliteNATBindMappingExtAddressType OBJECT-TYPE
          SYNTAX InetAddressType
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
          "Type of the mapping’s external address."
          ::= { dsliteNATBindEntry 3 }
```

```
     dsliteNATBindMappingExtAddress OBJECT-TYPE
          SYNTAX InetAddress (SIZE (4|16))
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
          "The mapping’s external address. If this is the undefined
          address, all external addresses are mapped to the internal
          address."
          ::= { dsliteNATBindEntry 4 }

      dsliteNATBindMappingExtPort OBJECT-TYPE
          SYNTAX InetPortNumber
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
          "The mapping’s external port number. If this is zero, all
          external ports are mapped to the internal port."
          ::= { dsliteNATBindEntry 5 }

       dsliteNATBindMappingIntRealm OBJECT-TYPE
          SYNTAX SnmpAdminString
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
          "The realm to which natMappingIntAddress belongs."
          ::= { dsliteNATBindEntry 6 }

      dsliteNATBindMappingIntAddressType OBJECT-TYPE
          SYNTAX InetAddressType
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
          "Type of the mapping’s internal address."
          ::= { dsliteNATBindEntry 7 }

      dsliteNATBindMappingIntAddress OBJECT-TYPE
          SYNTAX InetAddress
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
          "The mapping’s internal address. If this is the undefined
          address, addresses are not translated."
          ::= { dsliteNATBindEntry 8 }

      dsliteNATBindMappingIntPort OBJECT-TYPE
          SYNTAX InetPortNumber
          MAX-ACCESS read-only
          STATUS current
```

```
          DESCRIPTION
           "The mapping’s internal port number. If this is zero, ports
            are not translated."
           ::= { dsliteNATBindEntry 9 }

      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 10 }

      dsliteNATBindMappingMapBehavior OBJECT-TYPE
          SYNTAX NatBehaviorType
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
          "Mapping behavior as described in [RFC4787] section 4.1."
          ::= { dsliteNATBindEntry 11 }

      dsliteNATBindMappingFilterBehavior OBJECT-TYPE
          SYNTAX NatBehaviorType
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
          "Filtering behavior as described in [RFC4787] section 5."
          ::= { dsliteNATBindEntry 12 }

      dsliteNATBindMappingAddressPooling OBJECT-TYPE
          SYNTAX NatPoolingType
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
          "Type of address pooling behavior that was used to create
          this mapping."
          ::= { dsliteNATBindEntry 13 }

       dsliteNATBindMappingSubsIndex OBJECT-TYPE
           SYNTAX SubscriberIndex
           MAX-ACCESS read-only
           STATUS current
           DESCRIPTION
           "Subscriber using this mapping."
           ::= { dsliteNATBindEntry 14 }
```

```
--dsliteInfo

dsliteAFTRAlarmScalar OBJECT IDENTIFIER ::= { dsliteInfo 1 }


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

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

dsliteAFTRAlarmSpecificIP OBJECT-TYPE
   SYNTAX InetAddress
   MAX-ACCESS accessible-for-notify
   STATUS current
   DESCRIPTION
      " This object indicate the IP address whose port usage
       reach threshold "
   ::= { dsliteAFTRAlarmScalar 3 }

dsliteAFTRAlarmConnectNumber OBJECT-TYPE
   SYNTAX Integer32 (60..90)
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
      " This object indicate the threshold of DS-Lite
        connections alarm."
   ::= { dsliteAFTRAlarmScalar 4 }

dsliteStatisticTable OBJECT-TYPE
   SYNTAX SEQUENCE OF DsliteStatisticEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "This table provides statistical information
       of DS-Lite."
```

```
         ::= { dsliteInfo 2 }

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

      DsliteStatisticEntry ::=
         SEQUENCE {
       dsliteStatisticSubscriberIdex         SubscriberIndex,
       dsliteStatisticDiscard                Counter64,
       dsliteStatisticTransmitted            Counter64,
       dsliteStatisticIpv4Session            Counter64,
       dsliteStatisticIpv6Session            Counter64
      }


     dsliteStatisticSubscriberIdex OBJECT-TYPE
        SYNTAX SubscriberIndex
        MAX-ACCESS not-accessible
        STATUS current
       DESCRIPTION
           "Index of the subscriber or host.In ds-lite scenario
            This object indicate the address used for uniquely
            identifying the subscriber. It is the IPv6 address
            of B4 in DS-Lite."
         ::= { dsliteStatisticEntry 1 }

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

       dsliteStatisticTransmitted OBJECT-TYPE
         SYNTAX Counter64
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
            " This object indicate the number of packets received
```

```
              from or sent to this subscriber."
         ::= { dsliteStatisticEntry 3 }

      dsliteStatisticIpv4Session OBJECT-TYPE
         SYNTAX Counter64
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
            " This object indicate the number of the
              current IPv4 Session."
         ::= { dsliteStatisticEntry 4 }

      dsliteStatisticIpv6Session OBJECT-TYPE
         SYNTAX Counter64
         MAX-ACCESS read-only
         STATUS current
         DESCRIPTION
           " This object indicate the number of the
              current IPv6 Session."
        ::= { dsliteStatisticEntry 5 }

   ---dslite trap

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

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

        dsliteAFTRPortUsageOfSpecificIpAlarm NOTIFICATION-TYPE
         OBJECTS {  dsliteAFTRAlarmSpecificIP }
         STATUS current
         DESCRIPTION
            "This trap is triggered when used NAT
             ports of map address reach the threshold."
```

```
         ::= { dsliteTraps 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
             " Description the minimal requirements for conformance
               to the DS-Lite MIB."
          MODULE -- this module
              MANDATORY-GROUPS { dsliteNATBindGroup,
                  dsliteTunnelGroup,
                  dsliteStatisticGroup,
                  dsliteTrapsGroup,dsliteAFTRAlarmScalarGroup }
          ::= { dsliteCompliances 1 }

      dsliteNATBindGroup OBJECT-GROUP
         OBJECTS {
                  dsliteNATBindMappingIntRealm,
                  dsliteNATBindMappingIntAddressType,
                  dsliteNATBindMappingIntAddress,
                  dsliteNATBindMappingIntPort,
                  dsliteNATBindMappingPool,
                  dsliteNATBindMappingMapBehavior,
                  dsliteNATBindMappingFilterBehavior,
                  dsliteNATBindMappingAddressPooling }
         STATUS current
         DESCRIPTION
            " The collection of this objects are used to give the
             information about NAT Bind."
          ::= { dsliteGroups 1 }

      dsliteTunnelGroup OBJECT-GROUP
         OBJECTS { dsliteTunnelStartAddPreLen }
         STATUS current
         DESCRIPTION
            " The collection of this objects are used to give the
             information of tunnel in ds-lite."
         ::= { dsliteGroups 2 }
```

```
      dsliteStatisticGroup OBJECT-GROUP
         OBJECTS { dsliteStatisticDiscard,
                   dsliteStatisticTransmitted,
                   dsliteStatisticIpv4Session,
                   dsliteStatisticIpv6Session }
        STATUS current
        DESCRIPTION
           " The collection of this objects are used to give the
             statistical information of ds-lite."
          ::= { dsliteGroups 3 }

      dsliteTrapsGroup NOTIFICATION-GROUP
         NOTIFICATIONS { dsliteTunnelNumAlarm,
                         dsliteAFTRUserSessionNumAlarm,
                         dsliteAFTRPortUsageOfSpecificIpAlarm }
         STATUS current
         DESCRIPTION
            "The collection of this objects are used to give the
             trap information of ds-lite."
         ::= { dsliteGroups 4 }

       dsliteAFTRAlarmScalarGroup OBJECT-GROUP
         OBJECTS { dsliteAFTRAlarmB4Addr, dsliteAFTRAlarmProtocolType,
              dsliteAFTRAlarmSpecificIP,
             dsliteAFTRAlarmConnectNumber }
         STATUS current
         DESCRIPTION
            " The collection of this objects are used to give the
             information about AFTR alarming Scalar."
         ::= { dsliteGroups 5 }

             END
```

9. Security Considerations

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

Notification thresholds: An attacker setting an arbitrarily low treshold 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

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:

dsliteTunnelStartAddPreLen

dsliteNATBindMappingIntRealm

dsliteNATBindMappingIntAddressType

dsliteNATBindMappingIntAddress

dsliteNATBindMappingIntPort

dsliteNATBindMappingPool

dsliteNATBindMappingMapBehavior

dsliteNATBindMappingFilterBehavior

dsliteNATBindMappingAddressPooling

dsliteStatisticDiscard

dsliteStatisticTransmitted

dsliteStatisticIpv4Session

dsliteStatisticIpv6Session

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.

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 values recorded in the SMI Numbers registry, and the following IANA-assigned tunnelType values recorded in the IANAtunnelType-MIB registry:

```
Descriptor        OBJECT IDENTIFIER value
----------        -----------------------
DSLite-MIB         { transmission XXX }

IANAtunnelType ::= TEXTUAL-CONVENTION

            SYNTAX     INTEGER {

                       dsLite ("XX")        -- dslite tunnel

                       }
```

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

11. Acknowledgements

The authors would like to thanks the valuable comments made by Suresh Krishnan, Ian Farrer, Yiu Lee, Qi Sun, Yong Cui, Dave Thaler, Tassos Chatzithomaoglou and other members of SOFTWIRE WG.

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

12. References

12.1. Normative References

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

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

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

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

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

[RFC4008] Rohit, R., Srisuresh, P., Raghunarayan, R., Pai, N., and C. Wang, "Definitions of Managed Objects for Network Address Translators (NAT)", RFC 4008, March 2005.

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

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

12.2. Informative References

[I-D.ietf-behave-nat-mib] Perreault, S., Tsou, T., and S. Sivakumar, "Definitions of Managed Objects for Network Address Translators (NAT)", draft-ietf-behave-nat-mib-11 (work in progress), January 2014.

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

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

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

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

Authors' Addresses

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

Email: eleven.fuyu@huawei.com

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: Informational
Expires: October 21, 2014

Q. Sun
China Telecom
M. Chen
FreeBit
G. Chen
China Mobile
T. Tsou
Huawei Technologies
S. Perreault
Viagenie

April 19, 2014

Mapping of Address and Port (MAP) - Deployment Considerations
draft-ietf-softwire-map-deployment-04

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 October 21, 2014.

Table of Contents

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 not-yet-assigned or recalled 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.

```
                  +-------------------------------+
                  |        MAP Domain             |
              +---+---------------+--------------|
  +--------+  +                   |              |
  |        |  +---------+      +--+--+           |
  |  Host  |--|   CER   |      |     |           |
  |        |  |(MAP CE) |======| BNG | ======+---------+   +-----------+
  +--------+  +---------+      +--|--+       |  Core   |   |   IPv4    |
  +--------+      +---------------+          | Router  |---| Internet  |
  |        |  +---|-----+      +--+--+       |(MAP BR) |   |           |
  |  Host  |--|   CER   |======|     | ======+---------+   +-----------+
  |        |  |(MAP CE) |      | BNG |           |
  +--------+  +---------+      +--+--+           |
              +                   |              |
              +-------------------+--------------+
```

Figure 1: Stateless IPv4-over-IPv6 broadband access network architecture

When deploying MAP in home network, there are three network models as defined in [I-D.ietf-homenet-arch]: A. single ISP, single customer edge router (CER), internal routers; B. two ISPs, two CERs, shared

subnet; C. two ISPs, one CER, shared subnet. In MAP, the CE is usually deployed in CER, and the BR is deployed by an ISP. Models A and B are different from model C when using MAP. For models A and B, the CE (=CER) needs to correspond with only one MAP BR, while in model C one CE has to correspond with multiple MAP BRs. Figure 2 illustrates a typical case, where the home network has multiple connections to multiple providers or multiple logical connections to the same provider. In general, a CE may have different paths towards multiple MAP BRs.

```
          +-------+-------+     +-------+-------+         \
          |   Service     |     |   Service     |          \
          |  Provider A   |     |  Provider B   |           | Service
          |  Core Router  |     |  Core Router  |           | Provider
          |  ( MAP BR)    |     |  (MAP BR)     |           | network
          +-------+-------+     +-------+-------+           |
                   |                 |                     /
                   |    Customer     |                   /
                   |    Internet     |                  /
                   |   connections   |                |
                  +---------+---------+                 \
                  |       IPv6        |                   \
                  |   Customer Edge   |                    \
                  |      Router       |                    /
                  |    ( MAP CE)      |                   /
                  +---------+---------+                  /
                            |                           /
                            |                           | End-User
     ---+------------+-------+--------+-------------+--- | network(s)
        |            |                |             |     \
   +----+-----+ +----+-----+     +----+-----+ +-----+----+ \
   |IPv6 Host | |IPv6 Host |     | IPv6 Host| |IPv6 Host | /
   |          | |          |     |          | |          |/
   +----------+ +----------+     +----------+ +----------+
```

Figure 2: Relations between home networking and MAP domain

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. One MAP domain shares a common BR. All CEs in the MAP domain are provisioned with the same set of MAP rules by MAP DHCPv6 server [I-D.ietf-softwire-map-dhcp]. There might be multiple BMRs in one MAP domain, and CE would pick up its own BMR by longest prefix matching lookup. However, all CEs within the sub-domain will have the same BMR. In hub and spoke mode, CE would use DMR as its only FMR for outbound traffic; while in mesh mode, a longest-matching prefix lookup is done in the IPv4 routing table and the correct FMR is chosen.

[Note:Currently, there is no DMR in MAP-E. The IPv6 address of the BR could be provisioned by the DS-Lite AFTR Name option. But the DMR is still in use in MAP-T. Is this the final decision ?]

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 tak 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 recommanded to further divide the MAP domain into mutiple sub-domains, each with only one IPv4 subnet. This can simplify the MAP domain planning. 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 using the DMR as FMR, the packets must go though BR before arriving at the destination.

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 Figure 3 shown,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).

```
+---------------+---------+---------+---------+
|               |   CE1   |   CE2   |   CE3   |
+---------------+---------+---------+---------+
|      BMR      | rule 1  | rule 2  | rule 3  |
+---------------+---------+---------+---------+
|               | rule 1  | rule 1  | rule 2  |
|     FMRs      | rule 2  | rule 2  | rule 3  |
|               |         | rule 3  |         |
+---------------+---------+---------+---------+
```

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 mutiple 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

For PSID provisioning, all CEs with the same BMR should have the same PSID length. 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 even though more than 1024 ports are excluded as a result, which is due to the effects of rounding. The value of 'a' should be made explicitly provisionable 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.

If the fragment happens in the IPv6 stack, then only the first fragement contains full IPv4 destination address so that BR cannot do the decapsulation well until all fragments has been received. This disables the funtionality of anycast BR. To prevent this problem, MAP require the fragmentation is done in the IPv4 stack to fit the IPv6 domain path MTU. MAP-T and 4rd has not this problem as every IPv6 packet contains the full IPv4 address embedded into the IPv6 address and end-point reassembly is ensured.

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. 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 1280 bytes.

It is important that fragments of a MAP packet sent according to the Default Mapping Rule be handled by the same BR. This can be a problem when using an anycast BR address and routing fluctuations cause fragments of a packet to be routed to multiple BRs.

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 Managment

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 delegatable prefixes may not be actually delegatable (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 contigious 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 contigious aggregated block; when condition (b) is not satisfied, one has to devide 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 realtively 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 equiped 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 acclerate 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 prefered 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.

- o 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.

- o 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.

- o ISP could allocate a dedicated IPv6 prefix for MAP deployment. The allocation could not only facilitate the differentiation between MAPed traffic and native IPv6 trafffic, 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-08 (work in progress), April 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-10 (work in progress), January 2014.

[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-07 (work in progress), March 2014.

[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-05 (work in progress), February 2014.

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

[RFC5342] Eastlake, D., "IANA Considerations and IETF Protocol Usage for IEEE 802 Parameters", RFC 5342, September 2008.

[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.

[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

[I-D.ietf-homenet-arch]
Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", draft-ietf-homenet-arch-13 (work in progress), March 2014.

[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.

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

Email: fibrib@gmail.com

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
Viagenie
246 Aberdeen
Quebec, QC G1R 2E1
Canada

Phone: +1 418 656 9254
Email: simon.perreault@viagenie.ca

Softwire
Internet-Draft
Intended status: Standards Track
Expires: October 8, 2014

Y. Cui
J. Dong
P. Wu
M. Xu
Tsinghua University
A. Yla-Jaaski
Aalto University
April 6, 2014

Softwire Mesh Management Information Base (MIB)
draft-ietf-softwire-mesh-mib-06

Abstract

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Softwire mesh framework RFC 5565 [RFC5565] is a tunneling mechanism that enables the connectivity between islands of IPv4 networks across a single IPv6 backbone and vice versa. In 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 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, 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] and the softwire mesh framework RFC 5565 [RFC5565].

4. Conventions

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

5. Structure of the MIB Module

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

5.1. The swmSupportedTunnelTable Subtree

Since the AFBR needs to negotiate with a BGP peer what kind of tunnel they will use, it should first announce the types of tunnels it supports. The swmSupportedTunnelTable subtree provides the information. According to section 4 of RFC 5512 [RFC5512], current softwire mesh tunnel types include IP-IP, GRE and L2TPv3.

5.2. The swmEncapsTable Subtree

The swmEncapsTable subtree provides softwire mesh NLRI-NH information 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].

5.3. 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.

5.4. The swmConformance Subtree

The subtree provides the conformance information of MIB objects.

6. Relationship to Other MIB Modules

6.1. Relationship to the IF-MIB

The Interfaces MIB [RFC2863] defines generic managed objects for managing interfaces. Each logical interface (physical or virtual) has an ifEntry. Tunnels are handled by creating logical interfaces (ifEntry). Being a tunnel, softwire mesh 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 software 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 sent to the virtual interface for decapsulation into IPv6. The ifOutUcastPkts counts the number of IPv4 packets, which are generated by encapsulating IPv6 packets sent to the virtual interface. Particularly, if these IPv6 packets need to be fragmented, tifOutUcastPkts counts the number of packets after fragmentation. Similar definitions apply to other counter objects in the ifTable.

6.2. Relationship to the IP Tunnel MIB

The IP Tunnel MIB [RFC4087] contains objects applicable to all IP tunnels, including softwire mesh. Meanwhile, the Softwire Mesh MIB extends the IP Tunnel MIB to further describe encapsulation-specific information.

Running a point to multi-point tunnel, it is necessary for a softwire mesh AFBR to maintain an encapsulation table, used 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.

6.3. 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].

7. Definitions

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

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, transmission  FROM SNMPv2-SMI

    OBJECT-GROUP, MODULE-COMPLIANCE             FROM SNMPv2-CONF

    InetAddress, InetAddressType, InetAddressPrefixLength    FROM INET-ADDRESS-M
IB

    ifIndex                     FROM IF-MIB

    IANAtunnelType              FROM IANAifType-MIB;

swmMIB MODULE-IDENTITY
    LAST-UPDATED "201404060000Z"        -- April 6, 2014
```

```
    ORGANIZATION "Softwire Working Group"
    CONTACT-INFO "

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

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

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

             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 (2014).  This version
          of this MIB module is part of RFC yyyy; see the RFC
          itself for full legal notices."

    -- RFC Ed.: please replace yyyy with actual RFC number & remove this note.


    REVISION    "201404060000Z"
    DESCRIPTION
           "The MIB module is defined for management of object in
            the Softwire mesh framework."
    ::= { transmission 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 the AFBR supports,
        such as MPLS, L2TPv3, GRE, and IP-in-IP. There is
        no restriction of tunnel type the Softwire mesh can use."
    ::= { 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 and control 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."
    ::= { 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."
    ::= { 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."
    ::= { 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."
    ::= { 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
```

8. Security Considerations

The swmMIB module can be used for configuration of certain objects, and anything that can be configured can be incorrectly configured, with potentially disastrous results. 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 the tables and objects and their sensitivity/vulnerability:

swmSupportedTunnelType
swmEncapsIIPDstType
swmEncapsIIPDst
swmBGPNeighborTunnelType

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.

9. IANA Considerations

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

```
        Descriptor        OBJECT IDENTIFIER value
        ----------        -----------------------
        swmMIB            { transmission XXX }

       IANAtunnelType ::= TEXTUAL-CONVENTION
           SYNTAX     INTEGER {

                      softwireMesh ("xx")        -- softwire Mesh tunnel

                      }
```

Editor’s Note (to be removed prior to publication): the IANA is requested to assign a value for "XXX" under the ’mib-2’ subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "XXX" (here and in the MIB module) with the assigned value and to remove this note.

10. Acknowledgements

The authors would like to thank Dave Thaler, Jean-Philippe Dionne, Qi Sun, Sheng Jiang, Yu Fu for their valuable comments.

11. References

11.1. Normative References

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

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

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

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

[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, December 2002.

[RFC3826] Blumenthal, U., Maino, F., and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, June 2004.

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

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

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

[RFC5591] Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", RFC 5591, June 2009.

[RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, June 2009.

[RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", RFC 6353, July 2011.

11.2. Informative References

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

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

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

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

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

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

Authors' Addresses

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

Phone: +86-10-6260-3059
EMail: yong@csnet1.cs.tsinghua.edu.cn

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

Phone: +86-10-6278-5822
EMail: dongjiang@csnet1.cs.tsinghua.edu.cn

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

Phone: +86-10-6278-5822
EMail: weapon@csnet1.cs.tsinghua.edu.cn

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: January 5, 2015

C. Liu
Q. Sun
J. Wu
Tsinghua University

July 4, 2014

Dynamic IPv4 Provisioning for Lightweight 4over6
draft-liu-softwire-lw4over6-dhcp-deployment-03

Abstract

Lightweight 4over6 [I-D.ietf-softwire-lw4over6] is an IPv4 over IPv6 hub and spoke mechanism that provides overlay IPv4 services in an IPv6-only access network. Provisioning IPv4 addresse and port set to customers is the core function of Lightweight 4over6 control plane. [I-D.ietf-softwire-lw4over6] illustrates how to use DHCPv6 for deterministic IPv4 provisioning. This document discusses how to provision IPv4 parameters by using dynamic IPv4 provisioning protocols such as DHCPv4 over DHCPv6 [I-D.ietf-dhc-dhcpv4-over-dhcpv6].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Lightweight 4over6 [I-D.ietf-softwire-lw4over6] provides IPv4 access over IPv6 network in hub-and-spoke softwire architecture. In Lightweight 4over6, each Lightweight B4 (lwB4) is assigned with a port-restricted public IPv4 address or a full public IPv4 address to be used for IPv4 communication. Provisioning IPv4 address, port set and other IPv4 parameters to lwB4 is the core function of the Lightweight 4over6 control plane. It can be achieved by several protocols, such as DHCPv6 [RFC3315] [I-D.ietf-softwire-map-dhcp], DHCPv4 over DHCPv6 [I-D.ietf-dhc-dhcpv4-over-dhcpv6] , and PCP [RFC6887].

[I-D.ietf-softwire-lw4over6] illustrates how to use DHCPv6 for deterministic IPv4 provisioning. The IPv4 address and port set id (PSID) are carried in DHCPv6 options through DHCPv6 queries. However, the deterministic IPv4 provisioning adds some restrictions for addressing and deployment: the IPv4 lease time needs to be bound to the IPv6 lease time; the IPv4 address and PSID need to be embedded into clients' /128 IPv6 address so the client can not use arbitrary

/128 IPv6 address.

It’s not necessary to pre-install the binding between IPv4/IPv6 addresses through using dynamic IPv4 provisioning protocols such as DHCPv4 and PCP. Since DHCPv4 is unable to work in native IPv6 network directly, DHCPv4 over DHCPv6 [I-D.ietf-dhc-dhcpv4-over-dhcpv6] is proposed to support DHCPv4 functionality in IPv6 network by transporting DHCPv4 messages over DHCPv6 message. [I-D.ietf-dhc-dynamic-shared-v4allocation] describes how to allocate port set to clients using DHCPv4 over DHCPv6. PCP [RFC6887] also supports dynamic address and ports provisioning. [I-D.ietf-pcp-port-set] describes how port set is allocated to the lwB4.

This document discusses how to deploy Lightweight 4over6 using dynamic IPv4 provisioning protocols as the IPv4 provisioning protocol and analyses the benefit of using dynamic provisioning methods.

2. Terminology

Terminology defined in [I-D.ietf-dhc-dhcpv4-over-dhcpv6] and [I-D.ietf-softwire-lw4over6] is used extensively in this document.

3. Dynamic Provisioning Considerations

[I-D.ietf-softwire-lw4over6] describes the behavior of lwB4 and lwAFTR using DHCPv6 as provisioning protocol. It is based on a pre-determined binding relationship between IPv6 prefix and IPv4 address + PSID. This section discusses the issues produced by deterministic provisioning and how they could be solved by dynamic IPv4 provisioning.

3.1. lwB4 IPv6 Addressing

In section 5.1 of [I-D.ietf-softwire-lw4over6], it requires the lwB4 to embed its IPv4 address and PSID into its tunnel source IPv6 address. The reason to do this is that the binding relationship between IPv4/IPv6 addresses is pre-determined before the lwB4 requires IPv4/IPv6 addresses. Although the ISP can decide which IPv6 prefix for lwB4 to use, it can not decide the lwB4’s IPv6 suffix before IPv6 provisioning actually happens. When a lwAFTR receives an downstream IPv4 packet from Internet, it needs to construct the /128 IPv6 address of the lwB4 as the tunnel destination address and then encapsulate the packet. However since the binding table in lwAFTR is generated before IPv4 provisioning, the lwB4’s IPv6 suffix is unknown by looking up the binding table. So it is solved by filling IPv4 address and PSID into IPv6 suffix.

When using dynamic IPv4 provisioning, the binding entry in lwAFTR is generated after IPv4 provisioning process. When lwAFTR is encapsulating a packet, it looks up the binding table, using the destination IPv4 address and port as index, to get the lwB4's /128 IPv6 address. Thus IPv4 address and PSID are not necessary to be filled into IPv6 suffix. There is no restriction on how to generate the lwB4's IPv6 address.

3.2. lwB4 Tunnel Source Address

In deterministic IPv4 provisioning, because lwB4's prefixes are pre-determined in lwAFTR's binding table, when lwB4 has multiple prefixes, it needs to perform a longest prefix match to select which prefix to be used as the tunnel source address. A hint prefix is given by DHCPv6 server to tell the lwB4 which prefix looks like the correct one. If the lwB4 chooses a different prefix to be used as the tunnel address, it leads to a tunnel communication error.

With dynamic IPv4 provisioning, the tunnel source address is decided by lwB4. lwB4 can choose any of its IPv6 address as long as the address is routable to the lwAFTR.

3.3. Working with SLAAC

In deterministic IPv4 provisioning, the lwB4's /64 IPv6 prefix is assigned by ISP, and its /64 IPv6 suffix is filled with its IPv4 address and PSID. If a ISP wants multiple lwB4s to use the same /64 prefix, these lwB4s will run SLAAC to get the prefix. In this case, the upstream router advertises the /64 prefix to multiple lwB4s, then the lwB4s require their own IPv4 address and PSID through DHCPv6 Information-request. However, if the DHCPv6 server is not the upstream next hop of lwB4, it can not get the lifetime of lwB4's IPv6 address, so the DHCPv6 server is unable to recycle the allocated IPv4 addresses. Dynamic IPv4 provisioning works well with SLAAC since the IPv4 lease time is independent from IPv6 address.

3.4. lwAFTR Binding Table Maintenance

With dynamic IPv4 provisioning protocol, the lwAFTR's binding table is maintained through IPv4 provisioning process. To update a binding entry, lwB4's IPv6 address (as tunnel address), IPv4 address and PSID are needed. The IPv4 address and PSID are given by the provisioning server (DHCP 4o6 server or PCP server). The lwB4 must provide its /128 IPv6 address. If the IPv4 provisioning is transported in IPv6 unicast packet, the IPv6 source address in packets from lwB4 is used. Otherwise, the lwB4 must provide the IPv6 address in the message body, e.g. in a DHCPv6 option [I-D.fsc-softwire-dhcp4o6-saddr-opt].

When lwAFTR is located in the path of the IPv4 provisioning process, it updates its binding table through the provisioning messages. It listens all provisioning messages from provisioning server to lwB4, and updates binding through valid DHCP ACK message and PCP response.

When lwAFTR is out of the provisioning path, the provisioning server must indicate the lwAFTR about the binding updates. There are several protocols that support this function, e.g. NETCONF. A standardized data model may be needed, but it’s out of the scope of this document.

4. Working with DHCPv4 over DHCPv6

This section describes how DHCPv4 over DHCPv6 is used for Lightweight 4over6 configuration. [I-D.perreault-softwire-lw4over6-pcp] discusses how PCP works with Lightweight 4over6. In the remaining of this section, "lwB4" without explicitly written as "stateless lwB4" will refer to stateful lwB4 that runs DHCPv4 over DHCPv6 for dynamic IPv4 provisioning.

4.1. IP Addressing

Before starting DHCPv4 over DHCPv6 to achieve IPv4 configuration, lwB4 MUST be configured with an IPv6 address. There’s no restrictions on how IPv6 address is provisioned. The configured IPv6 address is used for IPv6 tunnelling and DHCPv4 over DHCPv6 process. The address that lwB4 chooses MUST be routable to the DHCP 4o6 server.

The softwire provider is free to provide any IPv4 address for a lwB4. There’s no restrictions on IPv6/IPv4 addressing, e.g. scattered IPv4 addresses can be used, and IPv4 address/PSID are not embeded into IPv6 addres.

4.2. DHCPv6 Configuration

Before stateful lwB4 runs DHCPv4 over DHCPv6 to acquire IPv4 address and port set, lwB4 SHOULD run DHCPv6 to achieve the DHCP 4o6 server’s IPv6 address. The DHCPv6 server provides the DHCP 4o6 server’s IPv6 address by OPTION_DHCP4_O_DHCP6_SERVER as defined in [I-D.ietf-dhc-dhcpv4-over-dhcpv6]. lwB4 MUST NOT require OPTION_S46_CONT_LW in all its DHCPv6 requests, and SHOULD ignore any OPTION_S46_CONT_LW and its encapsulated options in all the DHCPv6 responses.

It is possible that both stateless and stateful lwB4 exist in the same domain, and a DHCPv6 server serves both types of lwB4. When the DHCPv6 server receives a DHCPv6 query, if OPTION_S46_CONT_LW is

present in ORO, the lwB4 is treated as a stateless lwB4 that is asking for OPTION_S46_V4V6BIND and OPTION_S46_PORTPARAMS, and the DHCPv6 server processs the request as defined [I-D.ietf-softwire-map-dhcp]. When OPTION_DHCP4_O_DHCP6_SERVER is present in ORO regardless of whether OPTION_S46_CONT_LW is present, the server returns a OPTION_DHCP4_O_DHCP6_SERVER in the response with DHCP 4o6 server's IPv6 address. Both stateful and stateless lwB4 MAY run DHCPv4 over DHCPv6 to acheive stateless IPv4 information (i.e. DHCPINFORM query). When OPTION_S46_CONT_LW is not present in ORO, the DHCPv6 server MUST NOT reply any OPTION_S46_BR, OPTION_S46_V4V6BIND and OPTION_S46_PORTPARAMS to the client. The OPTION_S46_BR SHOULD be provided by DHCP 4o6 server in lwB4's DHCPv4 over DHCPv6 queries.

4.3. DHCPv4 over DHCPv6 Function

The DHCPv4 over DHCPv6 function in lwB4 is disabled by default, and enabled by OPTION_DHCP4_O_DHCP6_SERVER in DHCPv6 server's response. Once enabled, lwB4 runs stateful DHCPv4 over DHCPv6 to acquire IPv4 address and port set. lwB4 provides one of its IPv6 address as IPv6 tunnel source address to the DHCP 4o6 server, and get the lwAFTR's tunnel address through DHCPv4 over DHCPv6. The DHCPv4 over DHCPv6 message flow is described in section 4 of [I-D.fsc-softwire-dhcp4o6-saddr-opt] and MUST be followed.

4.4. Port Set Consideration

lwB4 gets its PSID through DHCPv4 over DHCPv6 along with its IPv4 address. [I-D.ietf-dhc-dynamic-shared-v4allocation] describes how to provision PSID to lwB4 through DHCPv4 over DHCPv6.

When sending a DHCPDISCOVER message, lwB4 MUST include OPTION_V4_PORTPARAMS in the Parameter Request List. If the server decides to reply a port-restricted address, it MUST reply OPTION_V4_PORTPARAMS to lwB4. if the server decides to reply a full IPv4 address, it SHOULD NOT reply OPTION_V4_PORTPARAMS in the response. When lwB4 receives DHCPv4 over DHCPv6 response without OPTION_V4_PORTPARAMS, it configures itself with the full IPv4 address as regular DHCPv4 client does. When lwB4 receives a shared IPv4 address, the address is used for NAPT and MUST NOT be used to identify the lwB4.

4.5. lwAFTR Binding Table Maintenance

lwAFTR maintains its binding table as per section 6.1 of [I-D.ietf-softwire-lw4over6]. The binding table is synchronized with DHCPv4 over DHCPv6 process. The following DHCPv4 over DHCPv6 messages triggers binding table modification:

- o DHCPACK: Generated by DHCP server, triggers lwAFTR to add a new entry or modify an existing entry.

- o DHCPRELEASE: Generated by lwB4, triggers lwAFTR to delete an existing entry.

When a DHCPACK event received by lwAFTR, the lwAFTR looks up its binding table using the IPv4 address and PSID. If there is an existing entry found, the lwAFTR updates the lifetime and IPv6 address field of the entry; otherwise the lwAFTR creates a new entry accordingly.

When a DHCPRELEASE event received by lwAFTR, the lwAFTR looks up its binding table using the IPv6 address, IPv4 address and PSID. The lwAFTR deletes the entry either by removing it from the binding table or mark the lifetime field to an invalid value (e.g. 0).

5. Security Considerations

Security considerations in [I-D.ietf-softwire-lw4over6] and [I-D.ietf-dhc-dhcpv4-over-dhcpv6] should be considered.

6. IANA Considerations

This document does not include an IANA request.

7. References

7.1. Normative References

[I-D.fsc-softwire-dhcp4o6-saddr-opt]
Farrer, I., Sun, Q., and Y. Cui, "DHCPv4 over DHCPv6 Source Address Option", draft-fsc-softwire-dhcp4o6-saddr-opt-00 (work in progress), July 2014.

[I-D.ietf-dhc-dhcpv4-over-dhcpv6]
Sun, Q., Cui, Y., Siodelski, M., Krishnan, S., and I. Farrer, "DHCPv4 over DHCPv6 Transport", draft-ietf-dhc-dhcpv4-over-dhcpv6-09 (work in progress), June 2014.

[I-D.ietf-dhc-dynamic-shared-v4allocation]
Cui, Y., Qiong, Q., Farrer, I., Lee, Y., Sun, Q., and M. Boucadair, "Dynamic Allocation of Shared IPv4 Addresses", draft-ietf-dhc-dynamic-shared-v4allocation-01 (work in progress), July 2014.

[I-D.ietf-softwire-lw4over6]
Cui, Y., Qiong, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture", draft-ietf-softwire-lw4over6-10 (work in progress), June 2014.

7.2. Informative References

[I-D.ietf-pcp-port-set]
Qiong, Q., Boucadair, M., Sivakumar, S., Zhou, C., Tsou, T., and S. Perreault, "Port Control Protocol (PCP) Extension for Port Set Allocation", draft-ietf-pcp-port-set-05 (work in progress), May 2014.

[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-07 (work in progress), March 2014.

[I-D.perreault-softwire-lw4over6-pcp]
Xie, C., Perreault, S., and C. Zhou, "Provisioning Lightweight 4over6 (lw4o6) with the Port Control Protocol (PCP)", draft-perreault-softwire-lw4over6-pcp-00 (work in progress), June 2013.

[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.

[RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.

Authors' Addresses

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

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

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

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

Phone: +86-10-6278-5983
Email: jianping@cernet.edu.cn

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 29, 2014

B. Sarikaya
Huawei USA
H. Ji
China Telecom
June 27, 2014

Multicast Support for Mapping of Address and Port Protocol
draft-sarikaya-softwire-map-multicast-02

Abstract

This memo specifies MAP-E's multicast component so that IPv4 hosts can receive multicast data from IPv4 servers over an IPv6 network. In the encapsulation solution for encapsulation variant of Mapping of Address and Port (MAP), MAP-E, IGMP Proxy at the MAP-E Customer Edge router uses IPv4-in-IPv6 tunnel established by MAP-E to exchange IGMP messages to establish multicast state at MAP-E Border Relay so that MAP-E Border Relay can tunnel IPv4 multicast data to IPv4 hosts connected to MAP-E Customer Edge device. In the Translation Multicast solution for the translation variant of MAP, MAP-T and 4rd, IGMP messages are translated into MLD messages at the CE router which is IGMP/MLD Proxy and sent to the network in IPv6. MAP-T/4rd Border Relay does the reverse translation and joins IPv4 multicast group for MAP-T/4rd hosts. Border Relay as multicast router receives IPv4 multicast data and translates the packet into IPv6 multicast data and sends downstream on the multicast tree. Member CEs receive multicast data, translate it back to IPv4 and transmit to the hosts.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

With IPv4 address depletion on the horizon, many techniques are being standardized for IPv6 migration including Mapping of Address and Port (MAP) - Encapsulation, - Translation and 4rd [I-D.ietf-softwire-map], [I-D.ietf-softwire-map-t], [I-D.ietf-softwire-4rd]. MAP/4rd enables

IPv4 hosts to communicate with external hosts using IPv6 only ISP network. MAP/4rd Customer Edge (CE) device's LAN side is dual stack and WAN side is IPv6 only. CE tunnels/translates IPv4 packets received from the LAN side to 4rd Border Relays (BR). BRs have anycast IPv6 addresses and receive encapsulated/translated packets from CEs over a virtual interface. MAP/4rd operation is stateless. Packets are received/ sent independent of each other and no state needs to be maintained except for NAT44 operation on IPv4 packets received from the user.

It should be noted that there is no depletion problem for IPv4 address space allocated for any source multicast and source specific multicast [RFC3171]. This document is not motivated by the depletion of IPv4 multicast addresses.

MAP-E, MAP-T and 4rd are unicast only. They do not support multicast. In this document we specify how multicast from home IPv4 users can be supported in MAP-E (as well as MAP-T and 4rd).

In case of MAP-E we integrate the multicast solution into the MAP-E tunnel resulting in a multicast tunneling protocol. Multicast tunneling protocol has the advantage of not requiring multicast enabled IPv6 network between CE routers and MAP-E BRs.

When MAP-E CE router receives an IGMP join message to an Any-Source Multicast (ASM) [RFC1112] or Source-Specific Multicast (SSM) group [RFC4607], it sends an aggregated IGMP membership report message in the IPv4-in-IPv6 tunnel to the border relay. MAP-E BR joins the source in the multicast infrastructure and sends multicast data downstream to all member CEs in the IPv4-in-IPv6 tunnel. When a CE has no membership state, i.e. after all member hosts leave the group(s), its state with the BR expires and the CE can send the next join message in anycast. IPv4 multicast data received at the BR is tunneled to the member CE in IPv6 and CE decapsulates the packet and sends IPv4 multicast data packet to the member hosts.

In case IPv6 network is multicast enabled, MAP-T/4rd can provide multicast service to the hosts using MAP-T/4rd Multicast Translation based solution. A Multicast Translator can be used that receives IPv4 multicast group management messages in IGMP and generates corresponding IPv6 group management messages in MLD and sends them to IPv6 network towards MAP-T/4rd Border Relay. We use [I-D.ietf-softwire-map-t] or [I-D.ietf-softwire-4rd] for sending IPv4 multicast data in IPv6 to the CE routers. At MAP-T/4rd CE router another translator is needed to translate IPv6 multicast data into IPv4 multicast data.

It should be noted that if IPv6 network is multicast enabled the translation multicast solution presented in Section 6 can also be used for MAP-E.

In this document we address MAP-E (and MAP-T/4rd) multicast problem and propose two architectures: Multicast Tunneling and Multicast Translation based solutions. Section 2 is on terminology, Section 3 is on requirements, Section 4 is on architecture, Section 5 is on multicast tunneling protocol Section 6 is on multicast translation protocol, and Section 7 states security considerations.

2. Terminology

This document uses the terminology defined in [I-D.ietf-softwire-map], [I-D.ietf-softwire-map-t], [I-D.ietf-softwire-4rd], [RFC3810] and [RFC3376].

3. Requirements

This section states requirements on MAP-E, MAP-T and 4rd multicast support protocol.

IPv4 hosts connected to MAP-E, MAP-T and 4rd CE router MUST be able to join multicast groups in IPv4 and receive multicast data.

Any source multicast (ASM) SHOULD be supported and source specific multicast (SSM) MUST be supported.

In case of encapsulation solution, MAP-E CE MUST support IGMP Proxy as defined in [RFC4605]. MAP-E BR MUST support IGMP querier downstream and MAP-E BR may support PIM protocol or IGMP router upstream.

In case of translation solution, MAP-T and 4rd CE MUST support IGMP to MLD translation. MAP-T and 4rd CE MUST be MLD Proxy as defined in [RFC4605]. MAP-T and 4rd BR MUST support MLD Querier. MAP-T and 4rd BR MUST support join/leave operations in IPv4 multicast upstream.

4. Architecture

In MAP-E, MAP-T and 4rd, there are hosts (possibly IPv4/ IPv6 dual stack) served by MAP-E, MAP-T and 4rd Customer Edge device. CE is dual stack facing the hosts and IPv6 only facing the network or WAN side. MAP-E, MAP-T and 4rd CE may be local IPv4 Network Address and Port Translation (NAPT) box [RFC3022] by assigning private IPv4 addresses to the hosts. MAP-E, MAP-T and 4rd CEs in the same domain may use shared public IPv4 addresses on their WAN side and if they do they should avoid ports outside of the allocated port set for NAPT

operation. At the boundary of the network there is MAP-E, MAP-T and 4rd Border Relay. BR receives IPv4 packets tunneled in IPv6 from CE and decapsulates them and sends them out to IPv4 network.

Unicast MAP-E, MAP-T and 4rd are stateless except for the local NAPT at the CE. Each IPv4 packet sent by CE treated separately and different packets from the same CE may go to different BRs or CEs. CE encapsulates IPv4 packet in IPv6 with destination address set to BR address (usually anycast IPv6 address). BR receives the encapsulated packet and decapsulates and send it to IPv4 network. CEs are configured with Rule IPv4 Prefixes, Rule IPv6 Prefixes and with an BR IPv6 anycast address. BR receives IPv4 packets addressed to this ISP and from the destination address it extracts the destination host’s IPv4 address and uses this address as destination address and encapsulates the IPv4 packet in IPv6 and sends it to IPv6-only network.

4.1. Encapsulation Multicast Architecture

Encapsulation variant of MAP called MAP-E network lends itself easily to the Multicast Tunneling architecture. Dual stack hosts are connected to the Customer Edge router and it is multicast enabled. It is assumed that IPv6 only network is the unicast only network and that IPv6 multicast is not enabled or IPv6 multicast is partially enabled. At the boundary of the network MAP-E Border Relay is connected to the native multicast backbone infrastructure.

We place IGMP Proxy at the CE router. CE router serves all the connected hosts. For multicast traffic, CE Router uses MAP-E tunneling interface with MAP-E BR to send/receive IGMP messages using IPv4-in-IPv6 tunnel [RFC2473].

MAP-E BR is IGMP Router towards the CEs and it could be IGMP Router or PIM router upstream. A given relay and all CEs connected to it can be considered to be on a separate logical link. On this link, gateways and relay communicate using IPv4-in-IPv6 tunneling to transmit and receive multicast control messages for membership management and multicast data from the relay to the gateways.

All the elements of MAP-E multicast support system with tunneling are shown in Figure 1.

```
        Dual Stack Hosts                                                IPv4
               +----+                                                  Network
               | H1 |                    IPv6
               +----+      +-------+    only       +--------------+      +
               +----+      |  CE   |    network    |      BR      |
               | H2 |   ---| IGMP  |---         -- |IGMP  | IGMP  |     IPv6
               +----+      | Proxy |               |Router|  PIM  |    Network
               +----+      +-------+               +--------------+
               | H3 |
               +----+
```

Figure 1: Architecture of MAP-E Multicast Tunneling

4.2. MAP-T and 4rd Translation Architecture

In case IPv6 only network is multicast enabled, translation multicast architecture can be used. CE implements IGMP Proxy function [RFC4605] towards the LAN side and MLD Proxy on its WAN interface. IPv4 hosts send their join requests (IGMP Membership Report messages) to CE. CE as a MLD proxy sends aggregated MLD Report messages upstream towards BR. CE replies MLD membership query messages with MLD membership report messages based on IGMP membership state in the IGMP/MLD proxy.

BR is MLD querier on its WAN side. On its interface to IPv4 network BR may either have IGMP client or PIM. PIM being able to support both IPv4 and IPv6 multicast should be preferred. BR receives MLD join requests, extracts IPv4 multicast group address and then joins the group upstream, possibly by issuing a PIM join message.

IPv4 multicast data received by the BR as a leaf node in IPv4 multicast distribution tree is translated into IPv6 multicast data by the translator using [I-D.ietf-softwire-map-t], [I-D.ietf-softwire-4rd] and then sent downstream to the IPv6 part of the multicast tree to all downstream routers that are members. IPv6 data packet eventually gets to the CE. At the CE, a reverse [I-D.ietf-softwire-map-t], [I-D.ietf-softwire-4rd] operation takes place by the translator and then IPv4 multicast data packet is sent to the member hosts on the LAN interface. [I-D.ietf-softwire-map-t], [I-D.ietf-softwire-4rd] are modified to handle multicast addresses.

In order to support SSM, IGMPv3 MUST be supported by the host, CE and BR. For ASM, BR MUST be the Rendezvous Point (RP).

MAP-T and 4rd Translation Multicast solution uses the multicast 46 translator in not one but two places in the architecture: at the CE

router and at the Border Relay. IPv4 multicast data received at 4rd BR goes through a [I-D.ietf-softwire-4rd] header-mapping into IPv6 multicast data at the BR and another [I-D.ietf-softwire-4rd] header-mapping back to IPv4 multicast data at the CE router. Encapsulation variant of [I-D.ietf-softwire-4rd] is not used. In case of MAP-T, IPv4 data packet is translated using v4 to v6 header translation using multicast addresses instead of the mapping algorithm used in [I-D.ietf-softwire-map-t].

All the elements of MAP-T and 4rd translation-based multicast support system are shown in Figure 2.

```
  Dual Stack Hosts                                                IPv4
   +----+                                                       Network
   | H1 |      +----------+          IPv6       +-------------+
   |    |      |    CE    |                     |     BR      |
   +----+      |Translator|          only       | Translator  |
               |MAP-T/4rd |                     |  MAP-T/4rd  |
   +----+      |          |          network    |        |IGMP|     +
   | H2 |   ---|IGMP-MLD  |---------         -- |MLD     | or |    IPv6
   +----+      |    Proxy |                     |Querier |PIM |   Network
   +----+      +----------+                     +-------------+
   | H3 |
   +----+
```

Figure 2: Architecture of MAP-T and 4rd Translation Multicast

5. Encapsulation Multicast Operation

When a host (H1, H2 or H3 in Figure 1) wants to join an IPv4 multicast group G or (S,G), it sends an IGMP report (IGMPv3 report for a source-specific group) to CE router.

CE encapsulates IGMP report messages in IPv6 and sends it over the tunnel to BR in anycast. CE router uses BR’s anycast address this CE router is configured with. After CE receives unicast address of BR, it sends all subsequent IGMP messages for G or (S,G) in unicast.

BR (topologically closest to this CE router) receives the message, decapsulates it and then lets IGMP router to process it. On the upstream, an IGMP Join message is sent to subscribe group G or (S,G) or a PIMv4 Join message is sent if PIM is supported. BR establishes membership state for group G or (S,G). BR sends all related IGMP messages to this CE in unicast using IPv4-in-IPv6 tunneling.

CE now has BR’s unicast address which it uses to send all IGMP packets for group G for any source multicast or (S,G) for source

specific multicast. If CE receives multiple join messages for the same group G, CE sends an aggregated join message to BR.

If CE receives another join message for a different group G', (S',G') CE encapsulates it and sends it in anycast to the BR. This enables the use of multiple BRs that may be deployed as anchor points and makes downstream multicast data delivery more efficient.

A CE is required to assist in IGMP signaling and data forwarding between the hosts that it serves and the corresponding BRs that are handling the multicast group G or (S,G). CE must have IGMP Proxy for each upstream tunnel interface that has been established with the BR. The CE decides on the mapping of downstream links to a proxy instance connected to an upstream link to a BR based on the unicast source IPv6 address in the packets received from BR. Because of this BRs MUST use the unicast source IPv6 address in packets sent to CEs. Encapsulation at the CE is according to [RFC2473] with an IPv4 payload carrying IGMP messages.

On the reception of IGMP reports from the hosts, the CE must identify the corresponding proxy instance from the incoming interface and perform regular IGMP proxy operations of inserting, updating or removing multicast forwarding state on the incoming interface and will merge state updates into the IGMP proxy membership database. It will then send an aggregated Report via the upstream tunnel to the BR when the membership database changes.

On the reception of IGMP queries, the CE proxy instance will answer the Queries on behalf of all active downstream receivers maintained in its membership database. Queries sent by the BR do not force the CE to trigger corresponding messages immediately towards hosts.

BR acts as the default multicast querier for the corresponding CE. It implements the function of the designated multicast router or a further IGMP proxy. After BR receives IGMP Join message it adds the tunnel to the CE in its outgoing interface list for the group (G) or the source, group (S,G) that the host wants to join. BR establishes group-/source-specific multicast forwarding states at its corresponding downstream tunnel interfaces. Afterwards, BR maintains/removes these group-/source-specific multicast forwarding states. BR treats its tunnel interfaces as multicast-enabled downstream links, serving zero to many listening nodes. BR will send a join message upstream towards the source of the multicast group to build a multicast tree in the native multicast infrastructure and becomes a leaf node in the multicast tree.

BR will send any group management messages (IGMP Report or Query messages) downstream to specific CEs on the tunnel interface by encapsulating these IGMP messages in IPv6 using [RFC2473].

As for multicast data, the data packets from the source received at the BR will be replicated to all interfaces in it’s outgoing interface list as well as the tunnel outgoing interface for all member CEs. BR sends multicast data in IPv4-in-IPv6 tunnel to the CE with the data packet encapsulated. Encapsulation is according to [RFC2473] with an IPv4 payload.

CE receives Multicast Data message over the tunnel interface associated with the tunnel to BR. After decapsulation, multicast traffic arriving at the CE on an upstream interface will be forwarded according to the group-specific or source-specific forwarding states as acquired for each downstream interface within the IGMP proxy instance.

5.1. Encapsulation Interface Considerations

Legacy IPv4 in IPv6 tunneling is performed as in [RFC2473]. Packets upstream from CE carry only IGMP signaling messages and they are not expected to be subject to fragmentation. However packets downstream, i.e. multicast data to CE may be subject to fragmentation.

Source and destination addresses of IGMP messages in IPv4-in-IPv6 softwire from CE is as follows:

Source address of IPv6 header is CE IPv6 address, e.g. 2001:db8:0:1::1, destination address is BR anycast address, possibly shared of the MAP domain.

Source address of IGMP messages is CE’s IPv4 interface address, e.g. 192.0.0.2, destination address is the all-systems multicast address of 224.0.0.1 for IGMP Query, all IGMPv3-capable multicast routers of 224.0.0.22 for IGMPv3 Report, the multicast group specified in the Group Address field of the Report for IGMPv1 or IGMPv2 Report.

Source and destination addresses of IGMP messages in IPv4-in-IPv6 softwire from BR is as follows:

Source address of IPv6 header is BR’s unicast IPv6 address, e.g. 2001:db8:0:2::1, destination address is CE IPv6 address, e.g. 2001:db8:0:1::1.

Source address of IGMP messages is CE’s IPv4 interface address, e.g. 192.0.2.1, destination address is the all-systems multicast address of 224.0.0.1 for IGMP Query, all IGMPv3-capable multicast routers of

224.0.0.22 for IGMPv3 Report, the multicast group specified in the Group Address field of the Report for IGMPv1 or IGMPv2 Report.

Source and destination addresses of multicast data messages in IPv4-in-IPv6 softwire is as follows:

Source address of IPv6 header is BR IPv6 unicast address, e.g. 2001:db8:0:2::1, destination address is CE IPv6 address, e.g. 2001:db8:0:1::1.

Source address of IPv4 multicast data is unicast IPv4 address of the multicast source, e.g. the content provider, destination address is IPv4 multicast group address.

BR decapsulates datagrams carrying IGMP messages from CE’s and then IGMP/PIM router processing takes over. Network Address Translation (NAT) is not applied on IGMP messages.

5.2. Avalanche Problem Considerations

In Section 5 BR replicates the data packets from the source received to all outgoing interfaces for all member CEs. This replication (often called avalanche problem) can be very costly if there are very large number of downstream member CEs such as in IPTV application. Note that the avalanche problem is faced by all multicast solutions that use tunneling to bypass non-multicast enabled access network.

In multicast MAP-E, one approach that can be used is to deploy MAP-E BRs close to the user. BRs colocated at the access network gateway such as at the Border Network Gateway (BNG) could reduce the packet duplication bottleneck considerably.

In multicast MAP-E, another approach is to exploit multiple BRs that can be deployed in the network. MAP-E CE can use BR anycast address when sending an encapsulated upstream IGMP join request and then use the unicast source address of this BR in subsequent IGMP messages.

6. MAP-T and 4rd Translation Multicast Operation

In this section we specify how the host can subscribe and receive IPv4 multicast data from IPv4 content providers based on the architecture defined in Figure 2 in two parts: address translation and protocol translation. Translation details are given in Appendix A.

6.1. Address Translation

IPv4-only host, H1 will join IPv4 multicast group by sending IGMP Membership Report message upstream towards the IGMP Proxy in Figure 2. MLD Proxy first creates a synthesized IPv6 address of IPv4 multicast group address using IPv4-embedded IPv6 multicast address format [I-D.ietf-mboned-64-multicast-address-format]. ASM_MPREFIX64 for any source multicast groups and SSM_MPREFIX64 for source specific multicast groups are used. Both are /96 prefixes.

SSM_MPREFIX64 is set to ff3x:0:8000::/96, with ’x’ set to any valid scope. ASM_MPREFIX64 values are formed as shown in Figure 3. M bit MUST BE set to 1. "flgs" and "scop" fields are defined in [RFC3956]. The usage of the "rsv" bits is the same as defined in [RFC3306]. "sub-group-id" field MUST follow the recommendations specified in [RFC3306] if unicast-based prefix is used or the recommendations specified in [RFC3956] if embedded-RP is used. The default value is all zeros.

```
|   8    |  4 |  4 |  4 |             76                |    32    |
+--------+----+----+----+-------------------------------+----------+
|11111111|flgs|scop|rsvM|         sub-group-id          |v4 address|
+--------+----+----+----+---------------------------------------------+
```

Figure 3: ASM_MPREFIX64 Formation

Each translator in the upstream BR is assigned a unique ASM_MPREFIX64 prefix. CE (MLD Proxy in CE) can learn this value by means out of scope with this document. With this, CE can easily create an IPv6 multicast address from the IPv4 group address a.b.c.d that the host wants to join.

Source-Specific Multicast (SSM) can also be supported similar to the Any Source Multicast (ASM) described above. In case of SSM, IPv4 multicast addresses use 232.0.0.0/8 prefix. IPv6 SSM_MPREFIX64 is set to FF3x:0:8000::/96.

Since SSM translation requires a unique address for each IPv4 multicast source, an IPv6 unicast prefix must be configured to the translator in the upstream BR to represent IPv4 sources. This prefix is prepended to IPv4 source addresses in translated packets.

The join message from the host for the group ASM_MPREFIX64:a.b.c.d or SSM_MPREFIX64:a.b.c.d or an aggregate join message will be received by MLD querier at the BR. BR as multicast anchor checks the group address and recognizes ASM_MPREFIX64 or SSM_MPREFIX64 prefix. It next checks the last 32 bits is an IPv4 multicast address in range

224/8 - 239/8. If all checks succeed, IGMPv4 Client joins a.b.c.d using IGMP on its IPv4 interface.

Joining IPv4 groups can also be done using PIM since PIM supports both IPv4 and IPv6. The advantage of using PIM is that there is no need to enable IGMP support in neighboring IPv4 routers. The advantage of using IGMP is that IGMP is a simpler protocol and it is supported by a wider range of routers. The use of PIM or IGMP is left as an implementation choice.

6.2. Protocol Translation

The hosts will send their subscription requests for IPv4 multicast groups upstream to the default router, i.e. Costumer Edge device. After subscribing the group, the host can receive multicast data from the CE. The host implements IGMP protocol’s host part.

Customer Edge device is IGMP Proxy facing the LAN interface. After receiving the first IGMP Report message requesting subscription to an IPv4 multicast group, a.b.c.d, MLD Proxy in the CE’s WAN interface synthesizes an IPv6 multicast group address corresponding to a.b.c.d and sends an MLD Report message upstream to join the group.

When CE is a NAT or NAPT box assigning private IPv4 addresses to the hosts, IP Multicast requirements for a Network Address Translator (NAT) and a Network Address Port Translator (NAPT) stated in [RFC5135] apply to IGMP messages and IPv4 multicast data packets.

When MAP-T or 4rd BR receives IPv4 multicast data for an IPv4 group a.b.c.d it [I-D.ietf-softwire-4rd] translates/encapsulates IPv4 packet into IPv6 multicast packet and sends it to IPv6 synthesized address corresponding to a.b.c.d using ASM_MPREFIX64 or SSM_MPREFIX64. The header mapping described in [I-D.ietf-softwire-4rd] Section 4.2 (using Table 1) is used except for mapping the source and destination addresses. In this document we use the multicast address translation described in Section 6.1 and propose it as a complementary enhancement to the translation algorithm in [I-D.ietf-softwire-4rd].

The IP/ICMP translation translates IPv4 packets into IPv6 using minimum MTU size of 1280 bytes (Section 4.3 in [I-D.ietf-softwire-4rd]) but this can be changed for multicast. Path MTU discovery for multicast is possible in IPv6 so 4rd BR can perform path MTU discovery for each ASM group and use these values instead of 1280. For SSM, a different MTU value MUST be kept for each SSM channel. Because of this 8 bytes added by IPv6 fragment header in each data packet can be tolerated.

Since multicast address translation does not preserve checksum neutrality, [I-D.ietf-softwire-4rd] translator/encapsulator at 4rd BR must however modify the UDP checksum to replace the IPv4 addresses with the IPv6 source and destination addresses in the pseudo-header which consists of source address, destination address, protocol and UDP length fields before calculating the new checksum.

IPv6 multicast data must be translated back to IPv4 at the 4rd CE (e.g. using Table 2 in Section 4.3 of [I-D.ietf-softwire-4rd]). Such a task is much simpler than the translation at 4rd BR because IPv6 header is much simpler than IPv4 header and IPv4 link on the LAN side of 4rd CE is a local link. The packet is sent on the local link to IPv4 group address a.b.c.d for IPv6 group address of ASM_MPREFIX64:a.b.c.d or SSM_MPREFIX64:a.b.c.d.

In case an IPv4 multicast source sends multicast data with the don’t fragment (DF) flag set to 1, [I-D.ietf-softwire-4rd] header mapping sets the D bit in IPv6 fragment header before sending the packet downstream as in Fig. 3 in Section 4.3 of [I-D.ietf-softwire-4rd]. This feature of [I-D.ietf-softwire-4rd] preserves the semantics of DF flag at the BR and CE.

Because MAP-T/4rd is stateless, Multicast MAP-T/4rd should stay faithful to this as much as possible. Border Relay acts as the default multicast querier for all CEs that have established multicast communication with it. In order to keep a consistent multicast state between a CE and BR, CE MUST use the same IPv6 multicast prefixes (ASM_MPREFIX64/SSM_REFIX64) until the state becomes empty. After that point, the CE may obtain different values for these prefixes, effectively changing to a different 4rd BR.

6.3. Supporting IPv6 Multicast in MAP-T and 4rd Translation Multicast

IPv6 multicast can be supported natively since IPv6-only network is assumed to be multicast enabled. MAP-T or 4rd Customer Edge device has MLD Proxy function. Proxy operation for MLD [RFC3810] is described in [RFC4605].

CE receives MLD join requests from the hosts and then sends aggregated MLD Report messages upstream towards BR. No address or protocol translation is needed at the CE or at the BR. IPv6 Hosts in MAP-T or 4rd domain use any source multicast block FF0X [RFC4291] or source specific multicast block FF3X::8000:0-FF3X::FFFF:FFFF for dynamic allocation by a host [RFC4607], [RFC3307].

MAP-T or 4rd Border Relay is MLD querier. It serves all CEs downstream. After receiving an MLD join message, BR sends PIM join message upstream to join IPv6 multicast group. Multicast membership

database is maintained based on the aggregated Reports received from downstream interfaces in the multicast tree.

MAP-T or 4rd Border Relay is a Rendezvous Point (RP) for ASM groups. For SSM, BR MUST support MLDv2.

IPv6 multicast data received from the Single Source Multicast or Any Source Multicast sources are replicated according to the multicast membership database and the data packets are sent downstream on the multicast tree and eventually received by the CEs that have one of more members of this multicast group.

MLD Proxy in the CE receives multicast data then forwards the packet downstream. Each member host receives IPv6 multicast data packet from its Layer 2 interface.

6.4. Learning Multicast Prefixes for IPv4-embedded IPv6 Multicast Addresses

CE can be pre-configured with Multicast Prefix64 of ASM_MPREFIX64 and SSM_MPREFIX64 that are supported in their network. However automating this process is also desired.

A new router advertisement option, a Multicast ASM Translation Prefix option, can be defined for this purpose. The option contains IPv6 ASM multicast translation prefix, ASM_MPREFIX64. A new router advertisement option, a Multicast SSM Translation Prefix option, can be defined for this purpose. The option contains IPv6 SSM multicast prefix translation prefix SSM_MPREFIX64.

After the host gets the multicast prefixes, when an application in the host wishes to join an IPv4 multicast group the host MUST use ASM_MPREFIX64 or SSM_MPREFIX64 and then obtain the synthesized IPv6 group address before sending MLD join message.

Source-specific multicast (SSM) group membership message payloads in IGMPv3 and MLDv2 contain address literals and their translation requires another multicast translation prefix option. IPv4 source addresses in IGMPv3 Membership Report message are unicast addresses of IPv4 sources and they have to be translated into unicast IPv6 source addresses in MLDv2 Membership Report message. A new router advertisement option, a Multicast Translation Unicast Prefix option can be defined for this purpose. The option contains IPv6 unicast Network-Specific Prefix U_PREFIX64. The host can be configured by its default router using router advertisements containing the prefixes [I-D.sarikaya-softwire-6man-raoptions]. 64:ff9b::/96 is the global value called well-known prefix that is assigned to U_PREFIX64 [RFC6052]. Organization specific values called Network-Specific

Prefixes can also be used. Since multicast is potentially inter-domain, the use of well-known prefix for U_PREFIX64 is recommended.

Note that U_PREFIX64 is also used in multicast data packet address translation. Source-specific multicast source address in multicast data packets coming from SSM sources MUST be translated using U_PREFIX64.

7. Security Considerations

4rd Encapsulation Multicast control and data message security can be provided by the security architecture, mechanisms, and services described in [RFC4301], [RFC4302] and [RFC4303]. 4rd Translation Multicast control and data message security are as described in [RFC4607] for source specific multicast.

8. IANA Considerations

TBD.

9. Acknowledgements

TBD.

10. References

10.1. Normative References

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

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

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, August 1989.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, February 1997.

[RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, October 1999.

[RFC3171] Albanna, Z., Almeroth, K., Meyer, D., and M. Schipper, "IANA Guidelines for IPv4 Multicast Address Assignments", RFC 3171, August 2001.

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

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

[RFC3307] Haberman, B., "Allocation Guidelines for IPv6 Multicast Addresses", RFC 3307, August 2002.

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

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

[RFC2765] Nordmark, E., "Stateless IP/ICMP Translation Algorithm (SIIT)", RFC 2765, February 2000.

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

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

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

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC5135] Wing, D. and T. Eckert, "IP Multicast Requirements for a Network Address Translator (NAT) and a Network Address Port Translator (NAPT)", BCP 135, RFC 5135, February 2008.

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

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

[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-10 (work in progress), January 2014.

[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-05 (work in progress), February 2014.

[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-08 (work in progress), April 2014.

[I-D.ietf-mboned-64-multicast-address-format]
Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X., and M. Xu, "IPv6 Multicast Address With Embedded IPv4 Multicast Address", draft-ietf-mboned-64-multicast-address-format-05 (work in progress), April 2013.

10.2. Informative references

[RFC3306] Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6 Multicast Addresses", RFC 3306, August 2002.

[RFC3956] Savola, P. and B. Haberman, "Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address", RFC 3956, November 2004.

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

[I-D.sarikaya-softwire-6man-raoptions]
Sarikaya, B., "IPv6 RA Options for Translation Multicast Prefixes", draft-sarikaya-softwire-6man-raoptions-01 (work in progress), February 2013.

[I-D.perreault-mboned-igmp-mld-translation]
Perreault, S. and T. Tsou, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Translation ("IGMP/MLD Translation")", draft-perreault-mboned-igmp-mld-translation-01 (work in progress), April 2012.

Appendix A. Group Membership Message Translation Details

IGMP Report messages (IGMP type number 0x12 and 0x16, in IGMPv1 and IGMPv2 and 0x22 in IGMPv3) are translated into MLD Report messages (MLDv1 ICMPv6 type number 0x83 and MLDv2 type number 0x8f). IGMP Query message (IGMP type number 0x11) is translated into MLD Query message (ICMPv6 type number 0x82) [I-D.perreault-mboned-igmp-mld-translation].

Destination address in ASM, i.e. IGMPv1, IGMPv2 and MLDv1, is the multicast group address so the destination address in IGMP message is translated into the destination address in MLD message using [I-D.ietf-mboned-64-multicast-address-format].

Destination address in SSM, i.e. IGMPv3 and MLDv2 is translated as follows: it could be all nodes on link, which has the value of 224.0.0.1 (IGMPv3) and ff02::1 (MLDv2), all routers on link, which has the value of 224.0.0.2 (IGMPv3) and ff02::2 (MLDv2), all IGMP/MLD-capable routers on link, which has the value of 224.0.0.22 (IGMPv3) and ff02::16 (MLDv2).

Source address of MLD message that CE sends is set to link-local IPv6 address of CE’s WAN side interface. Source address of MLD message that BR sends is set to link-local IPv6 address of BR’s downstream interface.

Multicast Address or Group Address field in IGMP message payloads is translated using [I-D.ietf-mboned-64-multicast-address-format] as described above into the corresponding field in MLD message.

Source Address in IGMPv3 message payloads is translated using U_PREFIX64, the IPv6 unicast prefix to be used by SSM source. [RFC6052] defines in Section 2.3 the address translation algorithm of embedding an IPv4 source address and obtaining an IPv6 source address using a network specific prefix like U_PREFIX64. At the BR on its upstream interface or at the CE on its LAN interface, IPv4 addresses are extracted from the IPv4-embedded IPv6 addresses.

Maximum Response Time (MRT) field in IGMPv2 and IGMPv3 queries are translated into Maximum Response Delay (MRD) in MLDv1 and MLDv2 queries, respectively. In the corresponding MLD message, MRD is set

to 100 times the value of MRT. At the BR on its upstream interface or at the CE on its LAN interface, MRT value is obtained by dividing MRD into 100 and rounding it to the nearest integer.

IGMP messages are sent with a Router Alert IPv4 option [RFC2113]. The translated MLD message are sent with a Router Alert option in a Hop-By-Hop IPv6 extension header [RFC2711]. In both cases, 2-octet value is set to 0.

Authors' Addresses

Behcet Sarikaya
Huawei USA
5340 Legacy Dr. Building 175
Plano, TX 75024

Email: sarikaya@ieee.org

Hui Ji
China Telecom
NO19.North Street
Beijing, Chaoyangmen,Dongcheng District
P.R. China

Email: jihui@chinatelecom.com.cn

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 15, 2014

Q. Sun
C. Xie
China Telecom
Y. Lee
Comcast
M. Chen
FreeBit

July 14, 2013

Deployment Considerations for Lightweight 4over6
draft-sun-softwire-lightweigh-4over6-deployment-04

Abstract

Lightweight 4over6 is a mechanism which moves the translation function from tunnel lwAFTR (AFTR) to lwB4s (B4s), and hence reduces the mapping scale on the lwAFTR to per-customer level. This document discusses various deployment models of Lightweight 4over6. It also describes the deployment considerations and applicability of the Lightweight 4over6 architecture.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 15, 2014.

Table of Contents

1. Introduction

Lightweight 4over6 [I-D.ietf-softwire-lw4over6] is an extension to DS-Lite which simplifies the AFTR module [RFC6333] with distributed NAT function among B4 elements. The lwB4 in Lightweight 4over6 is provisioned with an IPv6 address, an IPv4 address and a port-set. It performs NAPT on end user’s packets with the provisioned IPv4 address and port-set. IPv4 packets are forwarded between the lwB4 and the lwAFTR over a Softwire using IPv4-in-IPv6 encapsulation. The lwAFTR maintains one mapping entry per subscriber with the IPv6 address, IPv4 address and port-set. Therefore, this extension removes the NAT44 module from the AFTR and replaces the session-based NAT table to a per-subscriber based mapping table. This should relax the requirement to create dynamic session-based log entries. This mechanism preserves the dynamic feature of IPv4/IPv6 address binding as in DS-Lite, so it has no coupling between IPv6 address and IPv4 address/port-set as any full stateless solution ([RFC6052] or [I-D.ietf-softwire-map]) requires. This document discusses deployment models of Lightweight 4over6. It also describes the deployment considerations and applicability of the Lightweight 4over6 architecture.

Terminology of this document follows the definitions and abbreviations of [I-D.ietf-softwire-lw4over6].

2. Requirements Language

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

3. Deployment Model

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

Lightweight 4over6 can be deployed in a residential network (depicted in Figure1). In this scenario, a lwB4 would acquire an IPv4 address and a port-set after a successful user authentication process and IPv6 provisioning process. Then, it establishes an IPv4-in-IPv6 softwire using the IPv6 address to deliver IPv4 services to its connected host via the lwAFTR in the network. The lwB4 can act as a CPE, or software located in the host. The lwAFTR supports Lightweight 4over6 which keeps the mapping between lwB4's IPv6 address and its allocated IPv4 address + port set. The supporting system may keep the binding information as well for logging and user management.

```
                            +---------------+
                            |   Supporting  |
                            |    System     |
                            +-------+-------+
                                    |
                     +---------------+--------------|
                     |               |              |
+---------+  +------+---+        +--+--+           |
|  Host   |  |   lwB4   |        |     |           |
|         |--|          | ======-| BNG | === +---------+   +-----------+
+---------+  +----------+        +--|--+     |         |   |   IPv4    |
                                             | lwAFTR  |---| Internet  |
+---------+  +------+---+        +--+--+     |         |   |           |
|  Host   |--|   lwB4   | =======|     | ====+---------+   +-----------+
|         |  |          |        | BNG |           |
+---------+  +----------+        +--|--+           |
                     +               |              |
                     +---------------+--------------+
```

Figure 1 Deployment Model

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

IPv4 address is allocated to a given subscriber, the lwAFTR will report mapping records to the supporting system on creating a binding for traffic logging if necessary. Operators may use [I-D.ietf-behave-syslog-nat-logging] or [I-D.ietf-behave-ipfix-nat-logging] to report the port set allocated by lwAFTR. In this way, the lwAFTR can determine the binding by its own and there is little impact on existing network architecture. In top-down model, the Supporting system should firstly determine the binding information for each subscriber and then synchronize it with the lwAFTR. With this method, one binding record can be easily synchronized with multiple lwAFTRs and stateless failover can be achieved. However, new mechanism (e.g. [I-D.zhou-dime-4over6-provisioning]) needs to be introduced to notify each individual binding record between the Supporting system and the lwAFTR.

4. Overall Deployment Considerations

4.1. Addressing and Routing

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

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

4.2. Port-set Management

In Lightweight 4over6, each lwB4 will get its restricted IPv4 address and a port-set after successful user authentication process and IPv6 provisioning process. This port-set assignment can been achieved by DHCPv4-over-DHCPv6 [I-D.ietf-dhc-dhcpv4-over-dhcpv6] and PCP [I-D.ietf-pcp-port-set].

Operator may use DHCPv4 to provision IPv4 address to the lwB4. In a typical deployment, the DHCP server is a centralized DHCP server and lwAFTR is the DHCP relay agent to relay the dhcp messages to the server over unicast. Rarely DHCP server will collocate with the lwAFTR to provision IPv4 resources to the lwB4.

Operator may also use PCP Port-set Option to provision IPv4 address and port-set to the lwB4. In a typical deployment, PCP server will collocate with lwAFTR, and the subscriber’s binding can be determined by lwAFTR. The PCP request should be sent to the lwAFTR’s tunnel end-point address. It is not common that PCP server will be centralized deployed in which the lwAFTR is the PCP proxy to relay PCP requests.

It is also possible that subscriber’s binding is determined in AAA server. In this case, the BNGs will embed with a DHCPv4-over-DHCPv6 server function which allows them to locally handle any DHCPv4-over-DHCPv6 requests initiated by hosts. The AAA server will pass the subscriber’s binding to a BNG using the AAA attribute in [I-D.sun-softwire-lw4over6-radext] and in turn populates the mapping of the lwB4.

Some operators may offer different service level agreements (SLA) to users that some users may require more ports then others. In this deployment scenario, the operator can implement differentiated policies in provisioning system specified to a user’s lwB4 or a group of lwB4s to allocate a certain range of port-set. The lwAFTR may also run multiple instances with different port-set sizes to build the mapping table.

4.3. lwAFTR Discovery

A Lightweight 4over6 lwB4 must discover the lwAFTR’s IPv6 address before offering any IPv4 services. This IPv6 address can be learned through an out-of-band channel, static configuration, or dynamic configuration. In practice, Lightweight 4over6 lwB4 can use the same DHCPv6 option [RFC6334]to discover the FQDN of the lwAFTR.

When Lightweight 4over6 is deployed in the same place with DS-Lite, either different FQDNs can be configured for Lightweight 4over6 and DS-Lite separately or different DHCPv6 options can be used for Lightweight 4over6 [I-D.sun-softwire-lw4over6-dhcpv6] and DS-Lite. More detailed considerations on DS-Lite compatibility will be discussed in Section6.

4.4. Impacts on Accouting

In Lightweight 4over6, the accounting impact due to the tunneling protocol is the same with DS-Lite (see section 6.2 of [RFC6908]). However, since in Lightweight 4over6, the IPv4 service is only available after port-set allocation, if operators will regard IPv4 service as a on-demand value-added service, e.g. IPv6 connectivity is offered by default, while IPv4 connectivity will be offered untill a subscriber requires, etc., IPv4 service accounting should start after port-set allocation has completely.

5. lwAFTR Deployment Consideration

As Lightweight 4over6 is an extension to DS-Lite, both technologies share similar deployment considerations. For example: Interface consideration, Lawful Intercept Considerations, Blacklisting a shared IPv4 Address, AFTR’s Policies, AFTR Impacts on Accounting Process, etc., in [RFC6908] can also be applied here. This document only discusses new considerations specific to Lightweight 4over6.

5.1. Logging at the lwAFTR

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

Since the mapping in lwAFTR does not contain destination-specific information, operator should be aware that they will not be able to have destination-specific log.

5.2. MTU and Fragmentation Considerations

As Lightweight 4over6 is also a tunneling protocol, the same consideration regarding to the fragmentation and reassembly in DS-Lite [RFC6908] can also be applied. The only difference is that NAT functionality has been removed to lwB4 from lwAFTR in Lightweight 4over6. Therefore, on receiving an IPv4 fragmented packet after decapsulation in the lwB4, the lwB4 should further re-assemble the packets before doing NAT since the transport protocol information is only available in the first fragment.

5.3. Reliability Considerations of lwAFTR

Operators may deploy multiple lwAFTRs for robustness, reliability, and load balancing. In Lightweight 4over6, subscriber to IPv4 and port-set mapping must be pre-provisioned in the lwAFTR before providing IPv4 serives. For redundancy, the backup lwAFTR must either have the subscriber mapping already provisioned or notify the lwB4 to create a new mapping in the backup lwAFTR. The first option can be considered as Hot Standby mode, which requires state syncronization between multiple lwAFTRs. In Hot Standby mode, the bindings are replicated on-the-fly from the Primary lwAFTR to the Backup lwAFTR. When the Primary lwAFTR fails, the Backup lwAFTR will take over all the existing established sessions. In this mode, the internal hosts are not required to re-initiate the bindings with the

external hosts. In Lightweight 4over6, since the number of mapping states has been greatly reduced compared to DS-Lite, it is reasonable to adopt Hot Standby mode when there are only two lwAFTRs (one for Primary lwAFTR and one for Backup lwAFTR). However, if the number of lwAFTRs is larger than two, it is not scalable to deploy Hot Standby mode since each two of the lwAFTRs should to syncronize the binding states.

The second option is to use Cold Standby mode which does not require a Backup Standby lwAFTR to synchronize binding states. In failover, the lwAFTR has to notify the lwB4 to create a new binding, or fetch the binding by itself. [I-D.lee-softwire-lw4over6-failover] describes these two approaches for simple Cold Standby mode. For most deployment scenarios, we believe that Cold Standby mode should be sufficient enough and is thus recommended.

5.4. Placement of AFTR

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

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

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

5.5. Port set algorithm consideration

If each lwB4 is given a set of ports, port randomization algorithm can only select port in the given port-set. This may introduce security risk because hackers can make a more predictable guess of what port a subscriber may use. Therefore, non-continuous port set algorithms (e.g. as defined in [I-D.ietf-softwire-map]) can be used to improve security.

5.6. Path Consistency Consideration

In Lightweight 4over6, if the binding state is not syncronized among multiple lwAFTRs, the lwAFTR in which the subscriber’s binding state is stored should be exactly the one to service the subscriber. Otherwise, there will be no match in lwAFTR. This requires the provionsion packets (either using DHCPv4-over-DHCPv6 or PCP Port-set)

should arrive at the same lwAFTR as the subsquent IP-in-IP traffic. If multiple lwAFTRs are using the same Tunnel End Point address and there are intermediate routers between lwB4 and lwAFTR, there might be a problem when intermediate routers perform ECMP based on L4 hash for the plain provionsion packets while doing L3 hash for subsequent IP-in-IP traffic. In this case, it is recommended that the privioning packet is sent over IPv6 tunnel so that intermediate routers can only process ECMP using L3 hash.

6. lwB4 Deployment Consideration

For lwB4 consideration, the DNS Deployment Considerations and B4 Remote Management in [RFC6908] can also be applied here. In this section, we only describe the considerations sepcific to Lightweight 4over6.

6.1. NAT traversal issue

In Lightweight 4over6, since the subscriber’s source port will be restricted to the port-set allocated from the provisioning system, this will have impact on some NAT traversal mechanisms. For example, in UPnP 1.0, the external port number which can be used by remote peer is selected by UPnP client in end host. If the client randomly selects a port number which is not in that valid port-set, the UPnP process will fail. This is likely to happen because end-host does not know the port-set in lwB4. More detailed experimental results can be found in [I-D.deng-aplusp-experiment-results]. This problem will not exist in UPnP 2.0 because the UPnP client in the end-host will negotiate the external port number with the server. Another way is to implement a mechanism (e.g. [I-D.ietf-pcp-port-set], etc.) in end host to fetch the port-set from lwB4. The UPnP client can then select the port number within the port-set.

6.2. Static Port Forwarding Configuration

Currently, some external initiated applications rely on manual port configuration to reserve a port in the CPE. The restricted port-set in lwB4 will also have impacts on manual port forwarding configuration. It is recommended that the port-set allocated from the provioning system should be shown explicitly in the lwB4, which can be used as a hint for subscribers to add port forwarding mapping.

7. DS-Lite Compatibility Consideration

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

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

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

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

The first one is to distinguish using the client’s source IPv4 address. The IPv4 address from Lightweight 4over6 is public address as NAT has been done in the lwB4, and IPv4 address for DS-lite is private address as NAT will be done on AFTR. When the network element receives an encapsulated packet, it would de-capsulate packet and apply the transition mechanism based on the IPv4 source address in the packet. This requires the network element to examine every packet and may introduce significant extra load to the network element. However, both the B4 element and Lightweight 4over6 lwB4 can use the same DHCPv6 option [RFC6334] with the same FQDN of the AFTR and lwAFTR.

The second one is to distinguish using the destination’s tunnel IPv6 address. One network element can run separated instances for Lightweight 4over6 and DS-Lite with different tunnel addresses. Then B4 element and Lightweight 4over6 lwB4 can use the same DHCPv6 option [RFC6334] with different FQDNs pointing to corresponding tunnel addresses. This requires the supporting system should distinguish different types of users when assigning the FQDNs in DHCPv6 process. Another option is to use a new DHCPv6 option [I-D.sun-softwire-lw4over6-dhcpv6] to discover lwAFTR’s FQDN.

```
                    +---------------+--------------|
                    +               |              |
+---------+  +------+---+        +--+--+           |
```

```
|  Host   |  | LW 4over6|        |     |           |
|         |--| lwB4| ======-| BNG | === +-------------+   +-----------+
+---------+  +----------+        +--|--+     |LW 4over6    |   |   IPv4    |
                                             |lwAFTR/|---| Internet  |
+---------+  +------+---+        +--+--+     |DS-Lite AFTR |   |           |
|  Host   |--| DS-Lite  | =======|     | ====+-------------+   +-----------+
|         |  |    B4    |        | BNG |           |
+---------+  +----------+        +--|--+           |
                    +               |              |
                    +---------------+--------------+
```

Figure 2 DS-Lite Coexistence scenario with Integrated AFTR

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

This is similar to Case 1. The difference is the lwAFTR and AFTR functions won't be co-located in the same network element (depicted in Figure3). This use case decouples the functions to allow more flexible deployment. For example, an operator may deploy AFTR closer to the edge and lwAFTR closer to the core. Moreover, it does not require the network element to pre-configure with the CPE's IPv6 addresses. An operator can deploy more AFTR and lwAFTR at needed. However, this requires the B4 and lwB4 to discover the corresponding network element. In this case, B4 element and Lightweight 4over6 lwB4 can still use [RFC6334] with different FQDNs pointing to corresponding tunnel end-point addresses, and the supporting system should distinguish different types of users.

```
                    +---+---------------+-----------------|
                    +                   |                 |
+---------+  +------+---+        +------+-----+           |
|  Host   |  | LW 4over6|        |    BNG     |           |
|         |--| lwB4| ======-|DS-Lite AFTR| === +------------+   +-----------+
+---------+  +----------+        +------+-----+     |LW 4over6   |   |   IPv4
|
                                                    |lwAFTR|---| Internet  |
+---------+  +------+---+        +------+-----+     |            |   |
|
|  Host   |--| DS-Lite  | =======|    BNG     | ====+------------+   +-----------
+
|         |  |    B4    |        |DS-Lite AFTR|           |
+---------+  +----------+        +------+-----+           |
                    +                   |                 |
                    +-------------------+-----------------+
```

Figure 3 DS-Lite Coexistence scenario with Seperated AFTR

8. Acknowledgement

TBD

9. 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.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-11 (work in progress), February 2013.

[I-D.deng-aplusp-experiment-results]
Deng, X., Boucadair, M., and F. Telecom, "Implementing A+P in the provider’s IPv6-only network", draft-deng-aplusp-experiment-results-00 (work in progress), March 2011.

[I-D.ietf-behave-ipfix-nat-logging]
Sivakumar, S. and R. Penno, "IPFIX Information Elements for logging NAT Events", draft-ietf-behave-ipfix-nat-logging-00 (work in progress), March 2013.

[I-D.ietf-behave-syslog-nat-logging]
Chen, Z., Zhou, C., Tsou, T., and T. Taylor, "Syslog Format for NAT Logging", draft-ietf-behave-syslog-nat-logging-01 (work in progress), May 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-ipv6-06 (work in progress), March 2013.

[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-pcp-port-set]
Sun, Q., Boucadair, M., Sivakumar, S., Zhou, C., Tsou, T., and S. Perreault, "Port Control Protocol (PCP) Extension for Port Set Allocation", draft-ietf-pcp-port-set-01 (work in progress), May 2013.

[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-06 (work in progress), July 2013.

[I-D.ietf-softwire-dslite-deployment]
Lee, Y., Maglione, R., Williams, C., Jacquenet, C., and M. Boucadair, "Deployment Considerations for Dual-Stack Lite", draft-ietf-softwire-dslite-deployment-08 (work in progress), January 2013.

[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-lw4over6-00 (work in progress), April 2013.

[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-07 (work in progress), May 2013.

[I-D.lee-softwire-lw4over6-failover]
Lee, Y., Sun, Q., and C. Liu, "Simple Failover Mechanism for Lightweight 4over6", draft-lee-softwire-lw4over6-failover-00 (work in progress), July 2013.

[I-D.sun-softwire-lw4over6-dhcpv6]
Xie, C., Sun, Q., Lee, Y., Tsou, T., and P. Wu, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Lightweight 4over6", draft-sun-softwire-lw4over6-dhcpv6-00 (work in progress), July 2013.

[I-D.zhou-dime-4over6-provisioning]
Zhou, C. and T. Taylor, "Attribute-Value Pairs For Provisioning Customer Equipment Supporting IPv4-Over-IPv6 Transitional Solutions", draft-zhou-dime-4over6-provisioning-00 (work in progress), March 2013.

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

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

October 2010.

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

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

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

[RFC6908] Lee, Y., Maglione, R., Williams, C., Jacquenet, C., and M. Boucadair, "Deployment Considerations for Dual-Stack Lite", RFC 6908, March 2013.

1. Appendix:Experimental Result

We have deployed Lightweight 4over6 in our operational network of HuNan province, China. It is designed for broadband access network, and different versions of lwB4 have been implemented including a linksys box, a software client for windows XP, vista and Windows 7. It can be integrated with existing dial-up mechanisms such as PPPoE, etc. The major objectives listed below aimed to verify the functionality and performance of Lightweight 4over6:

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

1.1. Experimental environment

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

```
                                                +--------+
+-----+  +---------+                            | Syslog |
|Host1+--+lwB4|--+                         | Server |     --------
+-----+  +---------+  |                         +---+----+   ///       \\\
                      |         /------\            |       //           \\
                      |        //      \\    +---+----+ |               |
+-----+  +---------+  +-+--+  |   IPv6   |   |        | | IPv4 Internet |
|Host2+--|lwB4|--+BRAS+--|  Network |---| Concen-+-+               |
+-----+  +---------+  +-+--+   \\      //    | trator |  \\            //
                       |        \---+--/     +--------+   \\\        ///
                       |           |                        ---------
+-----+  +---------+   |           |
|Host3+--+lwB4+---+           |
+-----+  +---------+               |                         --------
                                  |                        //        \\
                                  |                       /           \
                                  +----------------------+IPv6 Internet +
                                                         |              |
                                                          \            /
                                                           \\        //
                                                             -------
```

Figure 2 Lightweight 4over6 experiment topology

In this deployment model, lwAFTR is co-located with a extended PCP server to assign restricted IPv4 address and port set for lwB4. It also triggers subscriber-based logging event to a centrilized syslog server. IPv6 address pools for subscribers have been distributed to

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

In our experiment, lwB4 will firstly get its IPv6 address and delegated prefix through PPPoE, and then initiate a PCP-extended request to get public IPv4 address and its valid port set. The lwAFTR will thus create a subscriber-based state accordingly, and notify syslog server with {IPv6 address, IPv4 address, port set, timestamp}.

1.2. Experimental results

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

The experimental results are listed as follows:

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

Figure 3 Lightweight 4over6 experimental result

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

1.3. Conclusions

From the experiment, we can have the following conclusions:

- Lightweight 4over6 has good scalability. As it is a lightweight solution which only maintains per-subscription state information, it can easily support a large amount of concurrent subscribers.

- Lightweight 4over6 can be deployed rapidly. There is no modification to existing addressing and routing system in our operational network. And it is simple to achieve traffic logging.

- Lightweight 4over6 can support a majority of current IPv4 applications.

Authors' Addresses

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

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

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

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

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

Email: yiu_lee@cable.comcast.com

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

Email: fibrib@gmail.com

Network Working Group
Internet-Draft
Intended status: BCP
Expires: November 6, 2014

S. Vinapamula
Juniper Networks
M. Boucadair
France Telecom

May 5, 2014

Recommendation for Prefix Binding in the Softwire DS-Lite Context
draft-vinapamula-softwire-dslite-prefix-binding-02

Abstract

This document discusses issues induced by the change of the Basic Bridging BroadBand (B4) IPv6 address and sketches a set of recommendations to solve those issues.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

IPv6 deployment models assumes IPv6 prefixes are delegated by Service Providers to the connected CPEs (Customer Premise Equipments) or hosts, which in their turn derive IPv6 addresses out of that prefix. In the case of DS-Lite [RFC6333], the Basic Bridging BroadBand (B4) element derives an IPv6 address for the softwire setup purposes.

A B4 element might obtain a new external IPv6 address, for a variety of reasons including a reboot of the CPE, power outage, DHCP lease expiry, or other action undertaken by the Service Provider. If this occurs, traffic forwarded to a B4’s previous address might be delivered to another B4 that now acquired that address. This affects all mapping types, whether implicit (e.g., by sending a TCP SYN) or explicit (e.g., using PCP [RFC6887]).

The main goal of this document is to propose recommendations to soften the impact of such renumbering issues.

Note that in some deployments, CPE renumbering may be required to accommodate some privacy-related requirements to avoid the same prefix be assigned to the same customer. It is out of scope of this document to discuss such contexts.

This document complements [RFC6908].

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].

3. The Problem

Since the network behind B4 can be overlapping across multiple CPEs, B4 address plays a key role in identifying associated resources assigned for each of the connections. These resources maintain state of Endpoint-Independent Mapping (EIM), Endpoint-Independent Filtering (EIF), Address Pooling Paired (APP) in the AFTR, and Port Control Protocol (PCP) mappings and flows.

However, there can be change in B4 address for any reason, may be because of change in CPE device or may be because of security extensions enabled in generating the IPv6 address. When the address change, the associated mappings created in the AFTR are no more valid. This may result in creation of new set of mappings.

Service Providers may want to limit the usage of these resources on

per subscriber basis for fairness resources usage. To that aim , a subscriber is identified by the delegated IPv6 prefix and not the derived B4 address. These policies are used for dimensioning purposes and also to ensure that AFTR resources are not exhausted. However when there is a change in B4 address, this policy doesn’t resolve stale mappings hanging around in the system, consuming not only system resources, but also reducing the available quota of resources per subscriber.

Clearing those mappings can be envisaged, but that will cause a lot of churn in the AFTR and could be disruptive to existing connections.

When services are hosted behind B4 element, and when there is a change in B4 address which if results in change in NAT address, these services have to advertise about their change, whenever there is a change of the B4 address. Means to discover the change of B4 address and NAT address is therefore required. Also, it doesn’t address latency issues where a service has to advertise its newly assigned external IP address and port and the clients have to consume and re-initiate connections.

PCP-specific failure scenarios are discussed in [I-D.boucadair-pcp-failure].

4. Recommendations

In order to mitigate the issues discussed in Section 3, the following recommendations are made:

1. A policy SHOULD be enforced at the AFTR level to limit the number of active softwires per subscriber. The default value MUST be 1. This policy aims to prevent a misbehaving subscriber to mount several softwires to consume more resources on the AFTR side.

2. Resource contexts created at the AFTR level SHOULD be based on the delegated IPv6 prefix and not based on the B4 address. Delegated prefix may be derived from the B4 address through a configured subscriber-mask. Administrators SHOULD configure per prefix limits of resource usage, instead of per tunnel limits. These resources include, number of flows, mappings including PCP, NAT pool resources, etc.

    1. Subscriber-mask is an integer that indicates the length of significant bits to be applied on the source IPv6 address (internal side) to identify a subscriber. Subscriber-mask is an AFTR system-wide configuration parameter that is used to enforce generic per-subscriber policies. Applying these generic policies does not require to configure every

subscriber prefix. Subscriber-mask must be configurable; the default value is 56.

2. For example, suppose an IPv6 prefix 2001::/56 is delegated to a CPE. Administrator should configure resource usage limits in AFTR based on the prefix 2001::/56 and not based on any B4 address derived from the delegated prefix. AFTR will derive the prefix from B4 address through configured subscriber-mask set to 56 by the administrator.

3. In the event a new IPv6 address is assigned to B4, the AFTR SHOULD migrate existing state to be bound to the new B4's IP address. This ensures the traffic destined to the previous B4 address will be redirected to the newer B4 address. The destination address for tunneling return traffic SHOULD be the last seen as B4's address from the CPE. Doing so avoids stale mappings and minimizes the risk of service disruption.

4. In the event of change of the CPE WAN's IPv6 prefix, unsolicited PCP ANNOUNCE messages are to be sent by the B4 element to internal hosts to update their mappings. This allows internal PCP clients to update their mappings with the new B4 IPv6 address and trigger updates to rendez-vous servers (e.g., dynamic DNS).

5. When a new prefix is assigned to the CPE, stale mappings may exist in the AFTR. This will consume both implicit and explicit resources. In order to avoid such issues, stable IPv6 prefix assignment is RECOMMENDED.

6. In case for any reason an IPv6 prefix has to be reassigned, it is RECOMMENDED to reassign an IPv6 prefix only when all the resources in use associated with that prefix are cleared from the AFTR. Doing so avoids to redirect traffic, destined to the previous prefix owner, to the new one.

5. Security Considerations

Security considerations related to DS-Lite are discussed in [RFC6333].

6. IANA Considerations

This document does not require any action from IANA.

7. Acknowledgements

G. Krishna and C. Jacquenet reviewed document and provided useful comments.

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.

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

[RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.

8.2. Informative references

[I-D.boucadair-pcp-failure] Boucadair, M. and R. Penno, "Analysis of Port Control Protocol (PCP) Failure Scenarios", draft-boucadair-pcp-failure-06 (work in progress), May 2013.

[RFC6908] Lee, Y., Maglione, R., Williams, C., Jacquenet, C., and M. Boucadair, "Deployment Considerations for Dual-Stack Lite", RFC 6908, March 2013.

Authors' Addresses

Suresh Vinapamula
Juniper Networks
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA

Phone: +1 408 936 5441
EMail: sureshk@juniper.net

Mohamed Boucadair
France Telecom
Rennes 35000
France

EMail: mohamed.boucadair@orange.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: December 28, 2014

B. Sarikaya
T. Tsou
Huawei Technologies (USA)
H. Ji
China Telecom
C. Zhou
Huawei Technologies
June 26, 2014

IPv6 Multicast in a 6rd Deployment
draft-zhou-softwire-6rdmulticast-01

Abstract

This memo specifies 6rd's multicast component so that IPv6 hosts can receive multicast data from IPv6 servers. In the 6rd encapsulation solution, multicast communication is completely integrated into the 6rd tunnel. In the 6rd translation solution, the protocol is based on proxying MLD at the 6rd Customer Edge router interworking the MLD messages to IGMP messages and sending them upstream through a network which supports IPv4 multicast. The 6rd Border Relay is a multicast router and interworks the IGMP to MLD for onward propasgation toward the IPv6 multicast source. IPv6 Multicast data received at 6rd Border Relay is translated into IPv4 multicast data and and delivered through the IPv4 multicast tree downstream to the 6rd Customer Edge. The latter translates it back to IPv6 multicast data then delivers it to the hosts.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

With IPv4 address depletion on the horizon, many techniques are being standardized for IPv6 migration including 6rd [RFC5969]. 6rd enables IPv6 hosts to communicate with external hosts using an IPv4-only legacy ISP network. The 6rd Customer Edge (CE) device's LAN side is dual stack and the WAN side is IPv4 only. The CE tunnels IPv6 packets received from the LAN side to 6rd Border Relays (BR) after encapsulating them as IPv4 packets. The BRs have anycast IPv4 addresses and receive encapsulated packets from CEs over a virtual

interface. 6rd operation is stateless. Packets are received/ sent independently of each other and no state needs to be maintained.

It should be noted that there is no depletion problem for IPv4 address space allocated for any source multicast and source specific multicast [RFC3171]. This document is not motivated by the depletion of IPv4 multicast addresses.

6rd as defined in [RFC5969] and [RFC5569] is unicast only. It does not support multicast. In this document we specify how multicast from home IPv6 users can be supported in 6rd. This is what is meant by 6rd multicast protocol.

In the 6rd encapsulation approach, 6rd multicast is integrated into the 6rd unicast solution. 6rd customer premise equipment (CPE) is extended to support an MLD proxy [RFC4605]. This proxy receives MLD Membership Report messages [RFC4601] requesting to join a multicast group from its subtended hosts. It tunnels aggregated join requests upstream to the 6rd Border Router (BR) using IPv6 in IPv4 encapsulation. The 6rd Border Router is extended to support an MLD querier, which sends join requests upstream towards the multicast source(s, becomes part of the multicast tree, and thus receives IPv6 multicast data. The 6rd Border Router encapsulates the IPv6 multicast data using 6rd’s IPv6 in IPv4 encapsulation and sends it to each member CPE that has joined the stream concerned. The CPE decapsulates the packet and the MLD proxy sends the IPv6 multicast data downstream to the member hosts.

In the translation approach, native IPv4 multicast support in the network between Customer Edge routers and Border Router can be exploited. The translation approach requires MLD to IGMP interworking at the Customer Edge and IGMP to MLD interworking at the border router. The border router needs to translate IPv6 multicast data into IPv4 multicast data and the Customer Edge router needs to translate IPv4 multicast data back into IPv6 multicast data.

6rd’s CE to CE forwarding feature is not used in either approach.

2. Terminology

This document uses the terminology defined in [RFC5969], [RFC5569], [RFC3810], [RFC3376], and [I-D.ietf-softwire-dslite-multicast].

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. Requirements

This section states requirements on 6rd multicast support protocol.

IPv6 hosts connected to 6rd CE router MUST be able to join multicast groups in IPv6 and receive multicast data.

Both any source multicast (ASM) and source specific multicast (SSM) MUST be supported.

6rd multicast MUST NOT introduce the need to use more IPv4 addresses, thereby contributing to public IPv4 address depletion.

4. Architecture

In 6rd, IPv6 or IPv4/IPv6 dual stack hosts are served by the 6rd Customer Edge device (CE). The CE is dual stack facing the hosts and IPv4 only facing the network or WAN side. The CE tunnels IPv6 packets in IPv4 to the 6rd Border Relay (BR). The BR decapsulates the tunneled packets and forwards them to the IPv6 network. In the reverse direction, the BR receives IPv6 packets from the IPv6 network tunnels them in IPv4 to the CE. The CE decapsulates the IPv6 packets and forwards them to the hosts.

Unicast 6rd is stateless. Each IPv6 packet sent by the CE is treated separately and different packets from the same CE may go to different BRs. The CE encapsulates IPv6 packets in IPv4 with the IPv4 destination address set to the BR address (usually an anycast IPv4 address). BRs are placed where IPv6 native connectivity exists to other networks. A CE is configured with its own IPv4 address (public or private), with a 6rd IPv6 prefix from which the CE’s IPv4 address can be derived, and with one or more BR IPv4 addresses. When the BR receives IPv6 packets addressed to the CE, it extracts the CE’s IPv4 address from the destination IPv6 address and uses this address as the destination address for the IPv4 encapsulation of the IPv6 packet. 6rd views the IPv4-only network as an NBMA link from the IPv6 point of view and all 6rd CEs and BRs are defined as off-link neighbors from one other.

4.1. 6rd Tunneling Architecture

In order to support multicast, the CE implements an MLD Proxy function [RFC4605]. IPv6 hosts send their join requests (MLD Membership Report messages) to CE. The CE as a proxy sends aggregated Report messages upstream towards BR in unicast using IPv6 in IPv4 encapsulation.

```
Dual Stack Hosts                                IPv4
+----+                                        Network
| H1 |                 IPv4
+----+      +-----+    only       +-------+     +
+----+      | CE  |    network -- |   BR  |
| H2 |   ---| MLD |--- IPv6       |   MLD |    IPv6
+----+      |Proxy|    in         |Querier|   Network
+----+      +-----+    IPv4       +-------+
| H3 |                Encapsulation
+----+
```

Figure 1: Architecture of 6rd Tunneling Multicast Protocol

The BR is the default multicast querier for the CE. The BR implements a multicast router function or it could be another MLD proxy.

All the elements of 6rd multicast support system are shown in Figure 1.

4.2. Translation Architecture

In order to support multicast, CE implements MLD Proxy [RFC4605] and MLD to IGMP interworking function [ID.perreault-igmp-mld-translation]. IPv6 hosts send their join requests (MLD Membership Report messages) to CE. CE as a proxy sends aggregated IGMP Report messages upstream towards BR.

In order to support SSM, MLDv2 [RFC3810] and IGMPv3 [RFC3376] must be supported by the CE and BR, and MLDv2 must be supported by the host.

The BR is the default multicast querier for the CE. The BR implements an IGMP to MLD interworking function and multicast router function or it could be another MLD proxy.

It is assumed that the IPv4 only network to which the CE and the BR are connected supports native IPv4 multicast.

All the elements of 6rd translation-based multicast support system are shown in Figure 2.

```
   Dual Stack Hosts                                              IPv4
   +----+                                                       Network
   | H1 |                            IPv4
   +----+      +----------------+    only   +------------------+
   +----+      | CE     MLD     |   network |IGMP   BR         |  +
   | H2 |   ---| MLD   IGMP     |-----------| MLD        MLD   |
   +----+      |Proxy Translator|           |Translator Querier|
   +----+      +----------------+           +------------------+
   | H3 |                                                       IPv6
   +----+                                                       Network
```

Figure 2: Architecture of 6rd Translation-Based Multicast

5. 6rd Tunneling Multicast Operation

In this section we specify how the host can subscribe and receive IPv6 multicast data from IPv6 content providers based on the architecture defined in Figure 1.

The hosts will send their subscription requests for IPv6 multicast groups upstream to the default router, i.e., Costumer Edge device. After subscribing the group, the host can receive multicast data from the CE. The host implements MLD protocol’s host part.

The Customer Edge device is an MLD Proxy. After receiving the first MLD Report message requesting subscription to an IPv6 multicast group, the CE establishes a tunnel interface with a Border Relay. The tunnel is IPv4 based but it will carry MLD messages back and forth and IPv6 multicast data messages downstream.

The CE is a regular MLD proxy and it keeps an MLD proxy membership database. The CE inserts multicast forwarding state on the incoming interface, and merges state updates into the MLD proxy membership database. The CE updates or remove elements from the database as required. The CE will then send an aggregated Report via the upstream tunnel to the BR when the membership database changes.

The CE answers MLD queries from the BR based on the membership database. The CE’s downstream link follows the traditional multipoint channel forwarding and does not pose any specific problems.

The CE receives IPv6 multicast data from the BR tunneled over the tunnel interface. The CE decapsulates the packet and then forwards it downstream. Each member host receives the data packet based on the Layer 2 multicast interface. No packet duplication is necessary.

The Border Relay acts as the default multicast querier for all CEs that have established an IPv4 tunnel with it. In order to keep a consistent multicast state between a CE and BR, once a CE is connected it will stay connected until the state becomes empty. After that point, the CE may establish another tunnel to a different BR.

According to aggregated MLD reports received from subtending CEs, the BR establishes group/source-specific multicast forwarding states at its corresponding downstream tunnel interfaces. After that, the BR maintains or removes the state as required by the aggregated reports received from CEs.

At the upstream interface, the BR procures for aggregated multicast membership maintenance. Based on the multicast-transparent operations of the CEs, the BR treats its tunnel interfaces as multicast enabled downstream links, serving zero to many listening nodes.

When the BR receives MLD join requests from downstream CEs, the BR sends PIM join messages upstream towards multicast source(s). This results in a multicast tree formation where the BR is at the leaf of the multicast tree, enabling the BR to receive IPv6 multicast data sent by the source.

Multicast traffic arriving at the BR is transparently forwarded according to its multicast forwarding information base. Multicast data is first replicated according to MLD multicast group state and then forwarded in IPv6-in-IPv4 tunnels from the BR to the corresponding CEs.

5.1. Tunnel Interface Considerations

IPv6 in IPv4 tunneling is performed as specified in [RFC4213]. Considerations specified in [RFC5969] apply. Packets passing upstream from the CE carry only MLD signaling messages and they are not expected to be fragmented. However packets downstream, i.e., multicast data to the CEs, may be subject to fragmentation.

Source and destination addresses of MLD messages in IPv6-in-IPv4 tunnel from CE are as follows:

- o The source address of IPv4 header is the CE WAN interface IPv4 address. The destination address is the BR anycast address when an invite message is sent to group G. Subsequent messages to group G contain the BR unicast address as destination address.

- o The source address of the inner MLD message is the link local address. The destination address is all MLDv2-capable multicast routers or FF02::16 for MLD Version 2 Multicast Listener Reports.

The source and destination addresses of MLD messages in the IPv6- in-IPv4 softwire from BR are as follows:

- o The source address of the IPv4 header is the BR IPv4 unicast address. The destination address is the CE IPv4 address. This also holds for multicast data.

- o The source address of the inner MLD message is the link local address. The destination address is the link-scope all-nodes multicast address (FF02::1) for General Queries, or the IPv6 multicast group address for specific queries.

The source address of IPv6 multicast data is the unicast IPv6 address of the multicast source, e.g., the content provider. The destination address is the3 IPv6 multicast group address.

5.2. Avalanche Problem

In Section 5.1, multicast data is replicated to all interfaces, i.e., to all member CEs at the BR. This replication (often called avalanche problem) can be very costly if is a very large number of downstream member CEs such as in the IPTV application. See Appendix A in [I-D.ietf-softwire-dslite-multicast].

In 6rd tunneling multicast, the avalanche problem can be reduced by careful network partitioning. More BRs can be deployed in areas where IPv6 users are increasing in numbers. Deploying BRs by collocating them with the access network gateway as with the Border Network Gateway (BNG) is another possibility.

In the 6rd tunneling multicast operation, CEs are enabled to exploit multiple BRs that can be deployed in the network by using the BR anycast address any time they send an upstream MLD join request and then using the same BR that received the join message in subsequent MLD messages by using the same BR’s unicast address.

6. 6rd Translation Multicast Operation

In this section we specify how the host can subscribe and receive IPv6 multicast data from IPv6 content providers based on the architecture defined in Figure 2.

The hosts will send their subscription requests for IPv6 multicast groups upstream to the default router, i.e., the Costumer Edge

device. After subscribing the group, the host can receive multicast data from the CE. The host implements the MLD protocol's host part.

The Customer Edge device is an MLD Proxy. After receiving the first MLD Report message requesting subscription to an IPv6 multicast group, the CE interworks the MLD Membership Report message to an IGMP Membership report message. It sends it upstream only if joining a new group is needed.

Address translation in generating an IGMP Membership report message is done as follows: the destination address is copied from the last 32 bits of IPv6 multicast group address. The CE inserts the IPv4 address of its WAN interface into the source address. It is assumed that the IPv6 multicast group address in MLD Report message conforms to the addressing scheme described in [I-D.ietf-mboned-64-multicast-address-format], for any-source and source-specific multicast address formats.

Source addresses in the MLDv2 payload are translated as follows. Multicast source addresses in MLD Membership Report message MUST use uPrefix64, i.e. 64:ff9b::/96 defined in [RFC6052]. uPrefix64 facilitates translation into an IPv4 source address to be used in IGMPv3 Membership Report messages for source-specific multicast, i.e., by extracting the last 32 bits of IPv6 source address.

The IGMP Report message is received by the IGMP Querier/Proxy upstream on the link. (Normally this node is the Broadband Network Gateway, BNG in broadband networks.) The IGMP Querier/Proxy sends IGMPv3 Report message to the neighboring routers to join the group. In networks where PIM is supported, the IGMP Report message may be received by the PIM Designated Router. The PIM router sends a PIMv4 join message to join an IPv4 group.

The border router that receives the join message translates the message into MLD. To join an IPv6 group for any-source multicast, the IPv6 Multicast group address is obtained from the destination address. For source-specific multicast, the IPv6 source address is generated after obtaining the IPv4 source address of Membership Report message's Group Record Source Address field. The BR sends the PIMv6 join message upstream towards the source.

The BR MUST act as the designated router to which the source of the source-specific IGMP join message is connected. The BR MUST act as the rendez-vous point (RP) of the multicast group for the any-source multicast IGMP join message. Normally there is one such BR in an operator's network. An IPv4 multicast tree eventually forms in the network between the CE and BR and an IPv6 multicast tree upstream from the BR for the same ASM or SSM group.

IPv6 multicast data received at the border router from the source is translated into IPv4. The last 32 bits of the source and destination address fields determine the source and destination addresses of the IPv4 multicast data packet. This packet is sent downstream on the multicast tree already formed for this IPv4 multicast group.

Multicast data packet address translation follows the rules in [I-D.ietf-mboned-64-multicast-address-format] for the multicast group address and [RFC6052] for source- specific multicast source address, i.e. using uPrefix64. For any- source multicast, the Border Router inserts an IPv4 source address, different for each source.

Packet header translation follows the rules in [RFC6145]. Fragmentation and reassembly are handled as described in [RFC6145]. After the IPv4 multicast data packet is sent downstream from the BR it may be fragmented by the routers.

The CE receives the IPv4 multicast data packet, possibly in fragments, and reassembles the fragments. The CE translates the IPv4 multicast data packet back to an IPv6 multicast data packet. Address translation is done following [I-D.ietf-mboned-64-multicast-address-format] for multicast group addresses and [RFC6052] for unicast SSM source addresses. Header translation is done as in [RFC6145].

IPv6 multicast data is sent on the home link to the host(s). IEEE 802.3 or IEEE 802.11 multicast link support usually handles this delivery in Layer 2 without any packet duplication if there are more than one members to the any-source multicast group or SSM source and multicast group.

6.1. Solution Based on Layer 2 Multicast Support

In this section we assume that Layer 2 multicast is supported in the network. Layer 2 multicast support is done in order to forward multicast data downstream to the ports of Layer 2 devices, i.e. switches that requested a multicast group instead of flooding the data to all the ports.

In the switches, called snooping switches, multicast MAC address based filters are set up which link layer 2 multicast groups to the egress ports. IGMP snooping switches are commonly used in operators' networks, most commonly at the access nodes (AN) [RFC6788].

When an IGMP Report message is received, the bridge will set up a multicast filter entry that allows (in case of a join message) or prevents (in case of a leave message) packets to flow on the port on which the IGMP Report message was received. In terms of IPv4

multicast addresses, the mapping is not unique as 32 IPv4 multicast addresses map to a single Ethernet multicast MAC address [RFC4541].

The main functionality of a snooping switch is to forward multicast data packets based on the filters that are setup, i.e. to those egress ports with multicast groups downstream and also to the router ports.

In a 6rd network the snooping switches must detect IGMP packets sent upstream by the CE and set the filtering rules accordingly. When IPv4 data packets are received the IGMP snooping switches forward these packets towards all CEs that have members, effectively achieving packet duplication at the access node level.

6.2. Analysis

An analysis of the translation solution reveals the following:\

- o The translation solution imposes a requirement on the IPv6 source-specific multicast sources to use uPrefix64 compatible source addresses. This requirement cannot be satified with simple configuration of the CPE router and Border Router.

- o In the case of any-source multicast, the border router must use a public IPv4 address distinctively to represent each IPv6 any-source multicast source.

- o In deployments which use IGMP routers, not PIM routers, source-specific multicast can be supported only if all routers have been upgraded to IGMPv3 and no IGMPv1 or IGMPv2 systems are present. Otherwise the operation reverts to the older version of IGMP to preserve compatibility and thus SSM can not be supported. With the use of PIM routers, this is avoided.

- o The border router must act as the designated router or the rendez-vous point for the IPv4/IPv6 multicast group and this may lead to the use of a single border router in the network instead of load sharing with various border routers.

7. Security Considerations

6rd Translation Multicast control and data message security are as described in [RFC5969]. The threats and their mitigation described in [RFC5969] apply to multicast communication as well.

8. IANA Considerations

TBD.

9. Acknowledgements

We would like to specially thank Mark Townsley for his constructive comments. Steve Wright’s online and very many offline comments helped us improve the document.

10. References

10.1. Normative References

[I-D.ietf-mboned-64-multicast-address-format]
Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X., and M. Xu, "IPv4-Embedded IPv6 Multicast Address Format (Work in progress)", August 2012.

[I-D.ietf-mboned-auto-multicast]
Bumgardner, G., "Automatic Multicast Tunneling (work in progress)", June 2012.

[I-D.ietf-softwire-dslite-multicast]
Qin, J., Boucadair, M., Jacquenet, C., Lee, Yiu., and Q. Wang, "Delivery of IPv4 Multicast Services to IPv4 Clients over an IPv6 Multicast Network", 2014.

[ID.perreault-igmp-mld-translation]
Perrault, S. and T. Tsou, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Translation ("IGMP/MLD Translation") (Work in progress)", February 2013.

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

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

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

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

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

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

[RFC4286] Haberman, B. and J. Martin, "Multicast Router Discovery", RFC 4286, December 2005.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.

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

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

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

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

[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.

10.2. Informative References

[RFC3171] Albanna, Z., Almeroth, K., Meyer, D., and M. Schipper, "IANA Guidelines for IPv4 Multicast Address Assignments", RFC 3171, August 2001.

[RFC6788] Krishnan, S., Kavanagh, A., Varga, B., Ooghe, S., and E. Nordmark, "The Line-Identification Option", RFC 6788, November 2012.

Authors' Addresses

B. Sarikaya
Huawei Technologies (USA)
5340 Legacy Dr. Building 175
Plano, TX 75024
USA

Email: sarikaya@ieee.org

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

Phone: +1 408 330 4424
Email: Tina.Tsou.Zouting@huawei.com
URI: http://tinatsou.weebly.com/contact.html

Hui Ji
China Telecom
NO19.North Street
Beijing, Chaoyangmen,Dongcheng District
P.R. China

Email: jihui@chinatelecom.com.cn

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

Email: cathy.zhou@huawei.com