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

Q. Wang
China Telecom
W. Meng
C. Wang
ZTE Corporation
M. Boucadair
France Telecom

June 3, 2015

RADIUS Extensions for IPv4-Embedded Multicast and Unicast IPv6 Prefixes

draft-hu-softwire-multicast-radius-ext-08

Abstract

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

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

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

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

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

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

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

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

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

2. Convention and Terminology

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

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

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

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

```
       mB4                                 NAS                    AAA
        |                                   |                    Server
        |------ DHCPv6 Solicit --------->   |                       |
        |                                   |                       |
        |                                   |----Access-Request ---->|
        |                                   |                       |
        |                                   |<---Access-Accept-------|
        |                                   |(Multicast-Prefixes-64) |
        |                                   |
                  |
        |<-------  DHCPv6 Advertise  ------|                        |
        |   (DHCPv6 OPTION_V6_PREFIX64 )    |                       |
        |                                   |                       |
        |-------  DHCPv6 Request  -------->|                        |
        |                                   |                       |
        |                                   |                       |
        |<----- DHCPv6 Reply -------------  |                       |
        |   (DHCPv6 OPTION_V6_PREFIX64 )    |                       |

                   DHCPv6                         RADIUS
```

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

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

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

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

```
      mB4                                NAS                 AAA
       |                                  |                 Server
       |                                  |                    |
       |----PPP LCP Config-Request------> |                    |
       |                                  |                    |
       |                                  |----Access-Request ---->|
       |                                  |                    |
       |                                  |<---- Access-Accept-----|
       |                                  | (Multicast-Prefixes-64)|
       |<-----PPP LCP Config-ACK  ------- |                    |
       |                                  |                    |
       |                                  |                    |
       |--- PPP IPv6CP Config-Request --->|                    |
       |                                  |                    |
       |<----- PPP IPv6CP Config-ACK -----|                    |
       |                                  |                    |
       |-------  DHCPv6 Solicit  -------->|                    |
       |                                  |                    |
       |<-------  DHCPv6 Advertise   -----|                    |
       |    (DHCPv6 OPTION_V6_PREFIX64 )  |                    |
       |                                  |                    |
       |-------  DHCPv6 Request  -------->|                    |
       |                                                  |
       |
       |                                  |                    |
       |<-------- DHCPv6 Reply ---------- |                    |
       |    (DHCPv6 OPTION_V6_PREFIX64)   |                    |

                 DHCPv6                         RADIUS
```

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

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

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

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

4. RADIUS Attribute

This section specifies the format of the new RADIUS attribute.

4.1. Multicast-Prefixes-64

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

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

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

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

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

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

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

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Length    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  asm-length   |               |
+-+-+-+-+-+-+-+-+               :
:    ASM_PREFIX64 (variable)    :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ssm-length   |               |
+-+-+-+-+-+-+-+-+               :
:    SSM_PREFIX64 (variable)    :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| unicast-length|               |
+-+-+-+-+-+-+-+-+               :
:     U_PREFIX64 (variable)     :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

Type:

145 for Multicast-Prefixes-64

Length:

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

asm-length:

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

ASM_PREFIX64:

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

ssm-length:

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

SSM_PREFIX64:

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

unicast-length:

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

U_PREFIX64:

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

5. Table of Attributes

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

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

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

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

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

6. Security Considerations

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

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

7. IANA Considerations

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

Multicast-Prefixes-64 - 145

8. Acknowledgments

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

9. Normative References

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

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

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

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

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

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

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

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

Authors' Addresses

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

Email: wangqian@ctbri.com.cn

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

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

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

Email: wang.cui1@zte.com.cn

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Network Working Group Q. Sun
Internet-Draft China Telecom
Intended status: Informational M. Chen
Expires: December 11, 2015 BBIX
G. Chen
China Mobile
T. Tsou
Huawei Technologies
S. Perreault
Jive Communications
June 9, 2015

Mapping of Address and Port (MAP) - Deployment Considerations
draft-ietf-softwire-map-deployment-06

Abstract

This document describes when and how an operator uses the technique of Mapping of Address and Port (MAP) for the IPv4 residual deployment in the IPv6-dominant domain.

Status of this Memo

Copyright Notice

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 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 can be two architecture: A. single ISP B. multihoming with two or more ISPs, sharing one CE. In the single ISP model, CE needs to communicate with only one MAP BR, while in multihoming model CE has to communicate with multiple MAP BRs. Figure 2 [RFC7368] illustrates a typical case, where the home network has multiple connections to multiple providers or multiple logical connections to the same provider. In the multihoming model, a CE will be provisioned with multiple BMRs. Routing information will also be configured for multihoming; but detail of the routing configuration is out of the scope of this memo.

```
           +-------+-------+     +-------+-------+         \
           |   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: MAP multihoming

4. Deployment Consideration

4.1. Building the MAP Domain

When deploying stateless MAP in an operational network, a provider should firstly do MAP domain planning based on that existing network. According to the definition of [I-D.ietf-softwire-map], a MAP domain is a set of MAP CEs and BRs connected to the same virtual link. All CEs in the MAP domain are provisioned with a same set of MAP rules by MAP DHCPv6 server [I-D.ietf-softwire-map-dhcp]. There might be multiple BMRs in one MAP domain, e.g. in case of multi-ISP. A CE may be provisioned with multiple IPv6 prefix, which can be used to find the corresponding BMR via longest prefix match. As defined in [I-D.ietf-softwire-map-dhcp], a BMR should be provisioned together with a BR IPv6 address; the CE should maintain this binding, so that the mapping between BMR and BR is achieved which is useful in multi-ISP scenario. In in mesh mode, a longest-matching prefix lookup is done in the IPv4 routing table and the correct FMR is chosen.

Basically, operator should firstly determine its own deployment topology for MAP domain as described in Section 4.1.1, as different considerations apply for different deployment models. Next, MAP domain planning, MAP rule provision, addressing and routing, etc., for a MAP domain should be taken into consideration, as discussed in the sections following Section 4.1.1.

For the scenario where one CE is corresponding with multiple MAP border relays, it is possible that those MAP BRs belong to different MAP domains. The CE must pick up its own MAP rules and domain parameters in each domain. This is a typical case of multihoming. The MAP rules must have the information about BR(s) and information about the service types and the ISP.

4.1.1. MAP Deployment Model Planning

In order to do MAP domain planning, an operator should firstly make the decision to choose mesh or hub and spoke topology according to the operator's network policy. In the hub and spoke topology, all traffic within the same MAP domain has to go through the BR, result in less optimal traffic flow; however, it simplifies CE processing since there is no need to do FMR lookup for each incoming packet. Moreover, it provides enhanced manageability as the BR can take full control of all the traffic. As a result, it is reasonable to deploy hub and spoke topology for a network with a relatively flat architecture.

In mesh topology, CE to CE traffic flows are optimized since they pass directly between the two nodes. Mesh topology is recommended when CE to CE traffic is high and there are not too many MAP rules, say fewer than 10 MAP rules, in the given domain.

4.1.2. MAP Domain Planning

Stateless MAP offers advantages in terms of scalability, high reliability, etc. As a result, it is reasonable to plan for a larger MAP domain to accommodate more subscribers with fewer BRs. Moreover, a larger MAP domain will also be easier for management and maintenance. However, a larger MAP domain may also result in less optimized traffic in the hub and spoke case, where all traffic has to go through a remote BR. In addition, it may result in an increased number of MAP rules and highly centralized address management. Choosing appropriate domain coverage requires the evaluation of tradeoffs.

When multiple IPv4 subnets are deployed in one MAP domain, it is recommended to further divide the MAP domain into mutiple sub-domains, each with only one IPv4 subnet. This can simplify the MAP domain planning. But there can be a side effect that it will increase the traffic between BRs. Different subdomains could be distinguished by different Rule IPv4 prefixes. As stated previously, all CEs within the same MAP subdomain will have the same Rule IPv4 prefix, Rule IPv6 prefix and PSID parameters.

4.1.3. MAP Rule Provisioning

In stateless MAP, Mesh or Hub and Spoke communications can be achieved among CEs in one MAP domain in terms of assigning appropriate FMR(s) to CEs. We recommend ISP deploy the full Hub and Spoke topology or full mesh topology describe below to simplify the configuration of the DHCPv6 server.

4.1.3.1. Full Hub and Spoke Communication among CEs

In order to achieve the full communication in the Hub and Spoke topology, no FMR is assigned to CEs. In this topology, when a CE sends packets to another CE in the same MAP domain via BR, or using the DMR as FMR, the packets must go though BR before arriving at the destination. DMR is specific for MAP-T only.

4.1.3.2. Full Mesh Communication among CEs

By assigning all BMRs in MAP domain to each CE as FMRs, Mesh communications can be achieved among all CEs. In this case, when CE receives an IPv4 packet, it looks up for an appropriate FMR with a specific Rule IPv4 prefix which has the longest match with the IPv4 destination address.

4.1.3.3. Mesh or Hub/Spoke communication among some CEs

Mesh communications among some CEs along with Hub/Spoke communications among some other CEs can be achieved by which differentiated FMRs are assigned to CEs. For instance, as shown in Figure 3, since both CE1 and CE2 has rule 1 and rule2, the communication between CE1 and CE2 can go directly without going though associated BR (Mesh topology). However, for CE1 and CE3, since there are no rule for each other, the communication between CE1 and CE3 must go though BR before reaching peer each other (Hub/Spoke topology).

	CE1	CE2	CE3
BMR	rule 1	rule 2	rule 3
FMRs	rule 1 rule 2	rule 1 rule 2 rule 3	rule 2 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 multiple BMRs, FMRs and DMRs.

MAP domain : DHCPv6 server = N:1 This might happen when MAP domain is relatively small and a single MAP DHCPv6 server is deployed in the network. In this case, multiple MAP domains should be distinguished based on CE's IPv6 prefix in different MAP domains.

4.1.5. PSID Consideration

If a provider would like to introduce differentiated address sharing ratios for different CEs, it is better to define multiple MAP sub-domains with different Rule IPv4 prefixes. In this way, MAP domain division is only a logical method, rather than a geographical one.

The default PSID offset(a) is chosen as 6 in [I-D.ietf-softwire-map] and this excludes the system ports (0-1023). For MAP, the initial part of the port number (the a-bits) cannot be zero (see Appendix B of [I-D.ietf-softwire-map].) As is shown in the section 3.2.4 of [I-D.tsou-softwire-port-set-algorithms-analysis], it is possible that a lower value of 'a' will give a higher sharing ratio and more than 1024 ports are excluded as a result, e.g. 'a' = 4 will exclude ports 0 - 4095. The value of 'a' should be made explicitly configurable by operators.

With regard to PSID format, both continuous and non-continuous port set can be supported in GMA algorithm. Non-continuous port set has the advantage of better UPnP friendly, while continuous port set is the simplest way to implement. Since PSID format should be supported not only in CPEs, BRs and DHCPv6 server, but also in other sustaining systems as well, e.g. traffic logging system, user management system, a provider should make the decision based on a comprehensive investigation on its demand and the capabilities of existing equipments.

Note that some ISPs may need to offer services in a MAP domain with a shared address, e.g. there are hosts FTP server under CEs. The service provisioning may require well-know port range (i.e. port range belong to 0-1023). MAP would provide operators with an option to generate a port range including those in 0-1023. Afterwards, operators could decide to assign it to any requesting user. However, if the port-set is too small, it is not suggested to assign one with only the port set 0~1023 or even less. Considerable non-well-known ports are surely needed. Another easier approach is assigning a dedicated IPv4 address to such a CE if the demand really exists.

4.1.6. Addressing and Routing

In MAP addressing, it should follow the MAP rule planning in the MAP domain.

For IPv4 addressing, since the number of scattered IPv4 address prefixes would be equal to the number of FMR rules within a MAP domain, one should choose as large IPv4 address pool as possible to reduce the number of FMR rules. For IPv6 address, the Rule IPv6 prefixes should be equal to the end user IPv6 prefix in MAP domain.

If ISP has a /24 rule IPv4 prefix with sharing ratio of 64 gives 16000 customers, and a /16 rule IPv4 prefix supports 4 million customer. If up the sharing ratio to 256, 64000 and 16 million customers can be supports respectively. For the ISP who has scattered IPv4 address prefixes, in order to reduce the number of FMRs, according to needs of ports they can divide different classes. For instance, for the enterprise customers class which need many ports to use, provision them the BMR with low sharing ratio while for the private customers class which don't need so many ports provision them the BMR with high sharing ratio.

For MAP routing, there are no IPv4 routes exported to IPv6 networks.

4.1.7. MAP vs. MAP-T vs. 4rd

Basically, encapsulation provides an architectural building block of virtual link where the underlay behavior is fully hidden, while translation does a delivery participating into the end-to-end transferring path where behaviors are exposed. It is reflected in the following aspects.

1. Option header

If translation or 4rd 'reversible translation' is applied, IPv4 options at the IP layer are not translated according to [RFC791][RFC2460], and packets with those options MUST be dropped by Domain-entry nodes, and return ICMPv4 error messages to signal IPv4-option incompatibility. This limitation is acceptable because there are a lot firewalls in current IPv4 Internet also filter IPv4 packets

2. ICMP

Some IPv4 ICMP codes do not have a corresponding codes in ICMPv6, a detailed analysis on the double translation behavior suggest that some ICMPv4 messages, when they are translated to ICMPv6 and back to ICMPv4 across the IPv6 domain, the accuracy might be sacrificed to some extent. Encapsulation keeps the full transparency of ICMPv4 messages.

Reversible translation approach of 4rd, however, does not translate ICMPv4 messages into ICMPv6 version. Instead, it treats ICMP as same as a transport layer protocol data unit. This behavior is similar to the encapsulation and keeps ICMP end-to-end transparency as well.

In either the encapsulation or translation mode, if an intermediate node generates an ICMPv6 error message, it should be converted into ICMPv4 version and returned to the source with a special source address and following the behavior specified in [RFC6791]. However, the behavior and semantics of the translation from ICMPv6 to ICMPv4 is different among encapsulation, translation and 4rd reversible translation approaches. Encapsulation treats routing error in the IPv6 domain as an (virtual)link error between the tunnel end points, while translation translate IPv6 routing error into corresponding IPv4 version, and 4rd, however, behaves according to whether the Tunnel Traffic Class option is set. The TTL behavior also reflect the differences among different approaches, which is worth paying attention to for the operating engineers. MAP-T translator is compatible with single translation approach.

3. PMTU and fragmentation

Both translation mode and encapsulation mode have PMTU and fragmentation problem. [RFC6145] discusses the problem in details for the translation, while [RFC2473] could be a reference on the issue in encapsulation.

4.2. BR Settings

1. BR placement

BR placement has important impacts on the operation of a MAP domain.

A first concern should be the avoidance of "triangle routing". In hub and spoke mode, all traffic will be routed through BR which may increase the path from the CE to an IPv4 peer. This can be accomplished easily by placing the BR close to the CE, such that the length of the path from the CE to the BR is minimized.

However, minimizing the CE-BR path would ignore a second concern, that of minimizing IPv4 operations. An ISP deploying MAP will probably want to focus on IPv6 operations, while keeping IPv4 operational expenditures to a minimum. This would imply that the size of the IPv4 network that the ISP has to administer would be kept to a minimum. Placing the BR near the CE means that the length of the IPv4 network between the BR and the IPv4 Internet would be longer.

Moreover, in case where the set of CEs is geographically dispersed, multiple BRs would be needed, which would further enlarge the IPv4 network that the ISP has to maintain.

Therefore, we offer the following guideline: BRs should be placed as close to the border with the IPv4 Internet as possible while keeping triangle routing to a minimum. Regional POPs should probably be considered as potential candidates.

Note also that MAP being stateless, asymmetric routing to/from the IPv4 Internet is natively supported and therefore no path-pinning mechanisms have to be additionally implemented.

Anycast can be used to let the network pick BR closest to a CE for traffic exiting the MAP domain. This is accomplished by provisioning a Default Mapping Rule containing an anycast IPv6 address or prefix. Operationally, this allows incremental deployment of BRs in strategic locations without modifying the provisioning system’s configuration. CE’s close to a newly-deployed BR will automatically start using it. The BR MUST participate in a dynamic IGP so that this can work automatically.

2. Reliability Considerations

Reliability of MAP is derived in major part from its statelessness. This means that MAP can benefit from the usual methods of Internet reliability.

Anycast, already mentioned in section 4.2.1, can be used to ensure reliability of traffic from CE to BR. Since there can be only one Default Mapping Rule per MAP domain, traffic from CE to BR will always use the same destination address. When this address is anycast, reliability is greatly increased. If a BR goes down, it stops advertising the IPv6 anycast address, and traffic is automatically re-routed to other BRs; the BR should also withdraw the routes for traffic from BR to CE, or the upstream routers connected to the BR should dynamically change the routes when it detects the failure of a BR, otherwise there will be a routing blackhole. For this mechanism to work correctly, it is crucial that the anycast route announcement be very closely tied to BR availability. See [RFC4786] for best current practices on the operation of anycast services. In practice, Equal-cost multi-path (ECMP) can be used to achieve active/active configuration. Operator can also increase the metric for one BR to have active/standby.

For reliability within a single link can be achieved with the help of a redundancy protocol such as VRRP [RFC5798]. This allows operation of a pair of BRs in active/standby configuration. No state needs to be shared for the operation of MAP, so there is no need to keep the standby node in a "warm" state: as long as it is up and ready to take over the virtual IPv6 address, quick failover can be achieved. This makes the pair behave as a single, much more reliable node, with less reliance on quick routing protocol convergence for reliability.

It is expected that production-quality MAP deployments will make use of both anycast and a redundancy protocol such as VRRP.

3. MTU/Fragmentation

If the MTU is well-managed such that the IPv6 MTU on the CE WAN side interface is set so that no fragmentation occurs within the boundary of the MAP domain, then the Tunnel MTU can be set to the known IPv6 MTU minus the size of the encapsulating IPv6 header (40 bytes). For example, if the IPv6 MTU is known to be 1500 bytes, the Tunnel MTU might be set to 1460 bytes. Without more specific information, the Tunnel MTU SHOULD default to 1240 bytes.

BRs using an anycast address as source can cause problems. If traffic sent by a BR with a source anycast address causes an ICMP error to be returned, that error packet’s destination address will be an anycast address, meaning that a different BR might receive it. In the case of a Too Big ICMP error, this could cause a path MTU discovery black hole. Another possible problem could occur if fragmented packets from different BRs using the same anycast address as source happen to contain the same fragment ID. This would break fragment reassembly. Since there is still no simple way to solve it completely, it is recommended to increase the MTU of the IPv6 network so that no fragmentation and Too Big ICMP error occurs.

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

On the other hand, in MAP domains where IPv4 addresses are shared, BRs and CEs may have to encapsulate or translate IPv4 packets whose IPv6 destinations depend on destination ports. Precautions are needed, due to the fact that the destination port of a fragmented datagram is available only in its first fragment. A sufficient precaution consists in reassembling each datagram received in multiple packets, and to treat it as though it would have been received in single packet. This function is such that MAP is in this case stateful at the IP layer. (This is common with DS-lite and NAT64/DNS64 which, in addition, are stateful at the transport layer.) At domain entrance, this ensures that all pieces of all received IPv4 datagrams go to the right IPv6 destinations.

4.3. CE Settings

1. bridging vs. routing

In routing manner, the CE runs a standard NAT44 [RFC3022] using the allocated public address as external IP and ports via DHCPv6 option. When receiving an IPv4 packet with private source address from its end hosts, it performs NAT44 function by translating the source address into public and selecting a port from the allocated port-set. Then it encapsulates/translates (depending on whether MAP-E or MAP-T is in use) the packet with the concentrator’s IPv6 address as destination IPv6 address, and forwards it to the concentrator. When receiving an IPv6 packet from the concentrator, the initiator decapsulates/translates the IPv6 packet to get the IPv4 packet with public destination IPv4 address. Then it performs NAPT44 function and translates the destination address into private one based on the entry in NAT state table in the CE.

The CE is responsible for performing ALG functions (e.g., SIP, FTP), as well as supporting NAT Traversal mechanisms (e.g., UPnP, NAT-PMP, manual mapping configuration). This is no different from the standard IPv4 NAT today.

For the bridging manner, end host would run a software performing CE functionalities. In this case, end host gets public address directly. It is also suggested that the host run a local NAT to map randomly generated ports into the restricted, valid port-set. Another solution is to have the IP stack to only assign ports within the restricted, valid range to applications. Either way the host guarantees that every source port number in the outgoing packets falls into the allocated port-set.

2. CE-initiated application

CE-initiated case is applied for situations where applications run on CE directly. If the application in CE use the public address directly, it might conflict with other CEs. So it is highly suggested that CE should also run a local NAT to map a private address to public address in CE. In this way, the CE IPv4 address passed to local applications would be conflict with other CEs.

4.4. Supporting System

1. Lawful Intercept

Sharing IPv4 addresses among multiple CEs is susceptible to issues related to lawful intercept. For details, see [RFC6269] section 12.

2. Traffic Logging

It is always possible for a service provider that operates a MAP domain to determine the IPv6 prefix associated with a MAP IPv4 address (and port number in case of a shared address). This mapping is static, and it is therefore unnecessary to log every IPv4 address assignment. However, changes in that static mapping, such as rule changes in the provisioning system, need to be logged in order to be able to know the mapping at any point in time.

Sharing IPv4 addresses among multiple CEs is susceptible to issues related to traffic logging. For details, see [RFC6269] sections 8 and 13.1.

3. Geo-location aware service

Sharing IPv4 addresses among multiple CEs is susceptible to issues related to geo-location. For details, see [RFC6269] section 7.

4. User Management

MAP IPv4 address assignment, and hence the IPv4 service itself, is tied to the IPv6 prefix lease; thus, the MAP service is also tied to this in terms of authorization, accounting, etc. For example, the MAP address has the same lifetime as its associated IPv6 prefix.

5. MAP Address Planning

This section is purposed to provide a referential guidance to operators, illustrating a common method of address planning with MAP in IPv4 residual deployment.

5.1. Planning for Residual Deployment, a Step-by-step Guide

Residual deployment starts from IPv6 address planning.

(A) IPv6 considerations

(A1) Determine the maximum number N of CEs to be supported, and, for generality, suppose N = 2^n.

For example, we suppose n = 20. It means there will be up to about one million CEs.

(A2) Choose the length x of IPv6 prefixes to be assigned to ordinary customers.

Consider we have a /32 IPv6 block, it is not a problem for the IPv6 deployment with the given number of CEs. Let x = 60, allowing subnets inside in each CE delegated networks.

(A3) Multiply N by a margin coefficient K, a power of two (K = 2 ^ k), to take into account that:

- Some privileged customers may be assigned IPv6 prefixes of length x’, shorter than x, to have larger addressing spaces than ordinary customers, both in IPv6 and IPv4;
- Due to the hierarchy of routable prefixes, many theoretically delegable prefixes may not be actually delegable (ref: host density ratio of [RFC3194]).

In our example, let’s take k = 0 for simplicity.

(B) IPv4 considerations

(B1) List all (non overlapping, not yet assigned to any in-running networks) IPv4 prefixes {Hi} that are available for IPv4 residual deployment.

Suppose that we hold two blocks and not yet assigned to any fixed network: 192.0.2.0/24 and 198.51.100.0/24.

(B2) Take enough of them, among the shortest ones, to get a total whose size M is a power of two (M = 2 ^ m), and includes a good proportion of the available IPv4 space.

If we use both blocks, M = 2^24 + 2^24, and therefore m = 25. Suppose the intended sharing ratio is 8 subscribers per address, resulting in (65536 - 1024)/8 = 8064 ports per subscriber assuming that the well-known ports are excluded. Then the PSID length to achieve this will be log2(8) = 3 bits. Bearing in mind the IPv4 24 bit prefix length for each of our two prefixes, the EA-bit length is (32 - 24) + 3 = 11 bits.

(B3) For each IPv4 prefix, Hi, of length hi, choose an prefix extension, say Ri of length ri = m - (32 - hi).

All these indexes must be non overlapping prefixes (e.g. 0, 10, 110, 111 for one /10, one /11, and two /12). In our example, we pick 0 for a contiguous address block while 1 for another.

Then we have:

```
H1 = 192.0.2.0/24, h1 = 24, r1 = 17 => R1 = bin(0);
H2 = 198.51.100.0/24, h2 = 24, r2 = 17 => R2 = bin(1);
```

Sometimes the IPv4 residual pool is not well aggregated and the contiguous address blocks may have different sizes. For example, in (B1), if we have H1 = 59.112.0.0/13 and H2 = 219.120.0.0/16 as the IPv4 residual pool, then M = 2^19 + 2^16, and in such a case, we must pick m so that m = ceil(log2(M)), where "ceil(x)" means the minimum integer not less than x, i.e., m = 20 in this case. Therefore r1 = 20 - (32 - 13) = 1, while r2 = 20 - (32 - 16) = 4. Several combinations are available for the R1 and R2 and one only needs to pay attention to avoiding overlapping when picking up the values.

(C) After (A) and (B), derive the rule(s)

(C1) Derive the length c of the MAP domain IPv6 prefix, C, that will appear at the beginning of all delegated prefixes (c = x - (n + k)).

(C2) Take any prefix for this C of length c that starts with a RIR-allocated IPv6 prefix.

(C3) For each IPv4 prefix Hi, make the rule, in which the key is Hi and the value is the domain IPv6 prefix C followed by the rule index Ri. Then this i-th rule’s Rule IPv6 Prefix will have the length of (c + ri).

Then we can do that:

```
c = 40 => C = 2001:0db8:ff00::/40
Rule 1: Rule IPv6 Prefix = 2001:0db8:ff00::/41
Rule 2: Rule IPv6 Prefix = 2001:0db8:ff80::/41
```

If we have different lengths for the Rule IPv4 prefix (as the extra example discussed at the end of (B)), their Rule IPv6 prefixes should not have the same length, as their rule index length is different.

As a result, for a certain CE delegating 2001:0db8:ff98:7650::/60, its parameters are:

```
Rule IPv6 Prefix = 2001:0db8:ff80::/41 => Rule 2
IPv4 Suffix = bin(111 0110 0)
                           PSID = bin(101) = 0x5
Rule IPv4 Prefix = 198.51.100.0/24
CE IPv4 Address = 198.51.100.236
```

If different sharing ratio is demanded, we may partition CEs into groups and do (A) and (B) for each group, determining the PSID length for them separately.

5.2. Remarks on Deployment Paradigms

1. IPv6 address planning in residual deployment is independent of the usage of the residual IPv4 addresses. The IPv4 address pool for "residual deployment" contains IPv4 addresses not yet allocated to customers/subscribers and/or those already recalled from ex-customers, re-programmed into relatively well-aggregated blocks.

2. It is recommended to have the number of rule entries as less as possible so that the merit of statelss deployment is reflected in practical performances. However, this effort is often constrained by the condition of an operator whether (a): it holds large-enough contiguous IPv4 address block(s) for the residual deployment, and (b): a short-enough IPv6 domain prefix so that the /64 delegation is easily satisfied even the EA-bits is quite long. When condition (a) is not satisfied, sub-domains have to be defined for each relatively small but contiguous aggregated block; when condition (b) is not satisfied, one has to divide the IPv4 aggregates into smaller blocks artificially in order to reduce the length of EA-bits. When we have good conditions fitting (a) and (b), it is NOT recommended to define short EA-bits with small length of IPv4 suffix (the value p) nor to increase the number of rule entries (also the number of sub-

domains) unless it really has to.

3. An extreme case is, when EA-bits contain the full IPv4 address while a full IPv4 address is assigned to a CE, i.e., o = p = 32, and q = 0, the MAP address format becomes almost equivalent to RFC6052-format [RFC6052] except the off-domain IPv4 peer's mapped IPv6 address. This frees the domain to distribute rules but the DMR. In such a case, IPv6 addressing is fully dependent of IPv4, which defers from the typical residual deployment case. MAP is mainly designed for residual deployment but also applied for the case of legacy IPv4 networks keeping communication with the IPv4 world over the IPv6 domain without renumbering, as long as the address planning doesn't matter.

4. Another extreme case is, when EA-bits' length becomes to zero, i.e., o = p = q = 0, a rule actually defines a correspondence between an IPv6 address and an IPv4 address (or a prefix), without any algorithmic correlation to each other. Using such a case in practice is not prohibited by the specification, but it is not recommended to deploy null EA-bits in large scale as the concern discussed in the above Remark 2, and as it has the limitation that the PSID must be null (q = 0) and therefore multiple CEs sharing a same IPv4 address is not supported here. It is recommended to apply Lightweight 4over6 [I-D.ietf-softwire-lw4over6], if a full de-correlation between IPv6 address and IPv4 address as well as port range is demanded.

5. A not-so-extreme case, p = 0, o = q, i.e., only PSID is applied for the EA-bits, is also a case possibly happening in practice. It also potentially generates a huge number of rules and therefore large-scale deployment of this case is not recommended either.

6. For operators who would like to utilize "some bits" of IPv6 address to do service identification, QoS differentiation, etc., it is recommended that these special-purpose bits should be embedded before the EA-bits so as to reduce the possibility of bit-conflict. However, it requires quite shorter IPv6 aggregate prefix of the operator. The bit-conflict is more likely to happen in this case if different domains have different Rule prefix lengths. Operators with this demand should pay attention to the impact on the domain rule planning.

6. Migration Methodology

6.1. Roadmap for MAP-based Solution

6.1.1. Start from Scratch

IPv6 deployment normally involves a step-wise approach where parts of the network should properly updated gradually. As IPv6 deployment progresses it may be simpler for operators to employ a single-version network, since deploying both IPv4 and IPv6 in parallel would cost more than IPv6-only network. Therefore switching to an IPv6-only network in relatively small scale will become more prevalent. Meanwhile, a significant part of network will still stay in IPv4 for long time, especially at early stage of IPv6 transition. There may not be enough public or private IPv4 addresses to support end-to-end network communication, without segmenting the network into small parts with sharing one IPv4 address space. That is a time to introduce MAP to bridge these IPv4 islands through IPv6 network.

6.1.2. Coexiting Phases

SP has various deployment strategy in the middle of transition. It’s foreseeable that IPv6 would likely coexist with IPv4 in a long period. The MAP deployment would also fit into the coexisting mode. To be specific, dual-stack technology is recommended in RFC6180 as the simplest deployment model to advance IPv6 deployment. MAP technology could get along well with native IPv6 connections and compatible with residual IPv4 networks. RFC6264 described a incremental transition approach in order to migrate networks to IPv6-only. DS-Lite is treated as a technology to accelerate the whole process. MAP can also take the same role to achieve a smooth transition.

6.1.3. Exit Strategy

The benefit of IPv6-only + MAP is that all IPv6 flows would go directly to the Internet, no need for encapsulation or translation. In this way, as more content providers and service are available over IPv6, the utilization on MAP CE and BR goes down since fewer destinations require MAP progressing. This way would advance IPv6, because it provides everyone incentives to use IPv6, and eventually the result is an pure IPv6 network with no need for IPv4. As more content providers and hosts equipped with IPv6 capabilities , the MAP utilization goes down until it is eventually not used at all when all content is IPv6. In this way, MAP has an "exit strategy". The corresponding solutions will leave the network in time.

6.2. Migration Mode

IPv4 Residual deployment is a interim phase during IPv6 migration. It would be beneficial to ISPs, if this phase is as short as possible since end-to-end IPv6 traversal is the really goals. When IPv6 is getting more and more mature, MAP would be retired in a natural way .

6.2.1. Passive Transition

Passive Transition is following IPv4 retirement law. In another word, MAP would always get along with IPv4, even all nodes is dual-stack capable. At a later stage of IPv6 migration, MAP can also be served for dual-stack hosts, which is sending traffic through the IPv4 stack. There is still a value for this approach because it could steer IPv4 traffic to IPv6 going through a MAP CE processing. When it comes the time ISP decide to turn off IPv4, MAP would be unnecessary due to IPv4 disappearance.

6.2.2. Active Transition

Active Transition is targeting to accelerate IPv4 exit and increase native IPv6 utilization. A desirable way deploying MAP is only providing IPv6 traversal ability to a IPv4-only host. However, MAP CE can not determine received traffic is send from a IPv4 node or a dual-stack node. In the latter case, IPv6 utilization is preferred for the most part . When a network evolves to a post-IPv6 era, it might be good for ISPs to consider to implement enforcement rules to help IPv6 migration.

- 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 traffic, but also clearly observe the change of MAP traffic. When the traffic is reducing for a while, ISP could close the MAP functionalities in some specific area. It would result networks to native IPv6-only capable.

7. IANA Considerations

This specification does not require any IANA actions.

8. Security Considerations

There are no new security considerations pertaining to this document.

9. Contributors

The members of the MAP design team are:

Congxiao Bao, Mohamed Boucadair, Gang Chen, Maoke Chen, Wojciech Dec, Xiaohong Deng, Remi Despres, Jouni Korhonen, Xing Li, Satoru Matsushima, Tomasz Mrugalski, Tetsuya Murakami, Jacni Qin, Qiong Sun, Tina Tsou, Dan Wing, Leaf Yeh, and Jan Zorz.

Thanks to Chunfa Sun who was an active co-author of some earlier versions of this draft. Thanks to Shishio Tsuchiya's valueable suggestion for this document.

10. Acknowledgements

Remi Despres contributed the original example of step-by-step deployment guidance in discussion with the authors. Ole Troan, as the head of MAP Design Team, joined the discussion directly and contributed a lot of ideas and comments. We also thank other members of the MAP Design Team for their comments and suggestions.

Thanks to Tom Talyer, Qi Sun and Ian Farrer for their thorough review and helpful comments.

11. References

11.1. Normative References

[I-D.ietf-softwire-4rd]
Despres, R., Jiang, S., Penno, R., Lee, Y., Chen, G., and M. Chen, "IPv4 Residual Deployment via IPv6 - a Stateless Solution (4rd)", draft-ietf-softwire-4rd-10 (work in progress), December 2014.

[I-D.ietf-softwire-map]
Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, "Mapping of Address and Port with Encapsulation (MAP)", draft-ietf-softwire-map-13 (work in progress), March 2015.

[I-D.ietf-softwire-map-dhcp]
Mrugalski, T., Troan, O., Farrer, I., Perreault, S., Dec, W., Bao, C., Yeh, L., and X. Deng, "DHCPv6 Options for configuration of Softwire Address and Port Mapped Clients", draft-ietf-softwire-map-dhcp-12 (work in progress), March 2015.

[I-D.ietf-softwire-map-t]
Li, X., Bao, C., Dec, W., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", draft-ietf-softwire-map-t-08 (work in progress), December 2014.

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

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

[RFC6791] Li, X., Bao, C., Wing, D., Vaithianathan, R., and G. Huston, "Stateless Source Address Mapping for ICMPv6 Packets", RFC 6791, November 2012.

11.2. Informative References

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

[RFC3194] Durand, A. and C. Huitema, "The H-Density Ratio for Address Assignment Efficiency An Update on the H ratio", RFC 3194, November 2001.

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

[RFC7368] Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", RFC 7368, October 2014.

Authors' Addresses

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

Phone: +86 10 5855 2923
Email: sunqiong@ctbri.com.cn

Maoke Chen
BBIX, Inc.
Tokyo Shiodome Building, Higashi-Shimbashi 1-9-1
Minato-ku, Tokyo 105-7310
Japan

Email: maoke@bbix.net

Gang Chen
China Mobile
28 Xuanwumenxi Ave; Xuanwu District
Beijing
P.R. China

Email: chengang@chinamobile.com

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

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

Simon Perreault
Jive Communications
Quebec, QC
Canada

Email: sperreault@jive.com

Softwire S. Jiang, Ed.
Internet-Draft Huawei Technologies Co., Ltd
Intended status: Standards Track Y. Fu, Ed.
Expires: December 16, 2019 CNNIC
C. Xie
China Telecom
T. Li
Tsinghua University
M. Boucadair, Ed.
Orange
June 14, 2019

RADIUS Attributes for Address plus Port (A+P) based Softwire Mechanisms
draft-ietf-softwire-map-radius-26

Abstract

IPv4-over-IPv6 transition mechanisms provide IPv4 connectivity services over IPv6 native networks during the IPv4/IPv6 co-existence period. DHCPv6 options have been defined for configuring clients for Lightweight 4over6, Mapping of Address and Port with Encapsulation, and Mapping of Address and Port using Translation unicast softwire mechanisms, and also multicast softwires. However, in many networks, configuration information is stored in an Authentication, Authorization, and Accounting server which utilizes the RADIUS protocol to provide centralized management for users. When a new transition mechanism is developed, new RADIUS attributes need to be defined correspondingly.

This document defines new RADIUS attributes to carry Address plus Port based softwire configuration parameters from an Authentication, Authorization, and Accounting server to a Broadband Network Gateway. Both unicast and multicast attributes are covered.

Status of This Memo

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 16, 2019.

Copyright Notice

Table of Contents

1. Introduction

Providers have started deploying and transitioning to IPv6. Several IPv4 service continuity mechanisms based on the Address plus Port (A+P) [RFC6346] have been proposed for providing unicast IPv4 over IPv6-only infrastructure, such as Mapping of Address and Port with Encapsulation (MAP-E) [RFC7597], Mapping of Address and Port using Translation (MAP-T) [RFC7599], and Lightweight 4over6 [RFC7596]. Also, [RFC8114] specifies a generic solution for the delivery of IPv4 multicast services to IPv4 clients over an IPv6 multicast network. For each of these mechanisms, DHCPv6 options have been specified for client configuration.

In many networks, user configuration information is stored in an Authentication, Authorization, and Accounting (AAA) server. AAA servers generally communicate using the Remote Authentication Dial In

User Service (RADIUS) [RFC2865] protocol. In a fixed broadband network, a Broadband Network Gateway (BNG) acts as the access gateway for users. That is, the BNG acts as both an AAA client to the AAA server, and a DHCPv6 server for DHCPv6 messages sent by clients. Throughout this document, the term BNG describes a device implementing both the AAA client and DHCPv6 server functions.

Since IPv4-in-IPv6 softwire configuration information is stored in an AAA server, and user configuration information is mainly transmitted through DHCPv6 between the BNGs and Customer Premises Equipment (CEs, a.k.a., CPE), new RADIUS attributes are needed to propagate the information from the AAA servers to BNGs so that they can be provided to CEs using the existing DHCPv6 options.

The RADIUS attributes defined in this document provide configuration to populate the corresponding DHCPv6 options for unicast and multicast softwire configuration, specifically:

o "Mapping of Address and Port with Encapsulation (MAP-E)" [RFC7597] (DHCPv6 options defined in [RFC7598]).

o "Mapping of Address and Port using Translation (MAP-T)" [RFC7599] (DHCPv6 options defined in [RFC7598]).

o "Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture" [RFC7596] (DHCPv6 options defined in [RFC7598]).

o "Unified IPv4-in-IPv6 Softwire Customer Premises Equipment (CPE): A DHCPv6-Based Prioritization Mechanism" [RFC8026].

o "Delivery of IPv4 Multicast Services to IPv4 Clients over an IPv6 Multicast Network" [RFC8114] (DHCPv6 options defined in [RFC8115]).

The contents of the attributes defined in this document have a 1:1 mapping into the fields of the various DHCPv6 options in [RFC7598], [RFC8026], and [RFC8115]. Table 1 shows how the DHCPv6 options map to the corresponding RADIUS attribute. For detailed mappings between each DHCPv6 option field and the corresponding RADIUS Attribute or field, see Appendix A.

DHCPv6 Option	RADIUS Attribute
OPTION_S46_RULE (89)	Softwire46-Rule
OPTION_S46_BR (90)	Softwire46-BR
OPTION_S46_DMR (91)	Softwire46-DMR
OPTION_S46_V4V6BIND (92)	Softwire46-V4V6Bind
OPTION_S46_PORTPARAMS (93)	Softwire46-PORTPARAMS
OPTION_S46_PRIORITY (111)	Softwire46-Priority
OPTION_V6_PREFIX64 (113)	Softwire46-Multicast

Table 1: Mapping between DHCPv6 Options and RADIUS Attributes

A RADIUS attribute for Dual-Stack Lite [RFC6333] is defined in [RFC6519].

This document targets deployments where a trusted relationship is in place between the RADIUS client and server.

2. Terminology

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

The reader should be familiar with the concepts and terms defined in [RFC7596], [RFC7597], [RFC7599], and [RFC8026].

The terms "multicast Basic Bridging BroadBand" element (mB4) and "multicast Address Family Transition Router" element (mAFTR) are defined in [RFC8114].

Softwire46 (S46) is used throughout to denote any of the IPv4-in-IPv6 softwire mechanisms listed above. Additionally, the following abbreviations are used within the document:

- BNG: Broadband Network Gateway
- BR: Border Relay
- CE: Customer Edge
- DMR: Default Mapping Rule
- lwAFTR: Lightweight AFTR

- o PSID: Port Set Identifier
- o TLV: Type, Length, Value
- o MAP-E: Mapping of Address and Port with Encapsulation
- o MAP-T: Mapping of Address and Port using Translation

3. New RADIUS Attributes

This section defines the following attributes:

1. Softwire46-Configuration Attribute (Section 3.1):

   This attribute carries the configuration information for MAP-E, MAP-T, and Lightweight 4over6. The configuration information for each Softwire46 mechanism is carried in the corresponding Softwire46 attributes. Different attributes are required for each Softwire46 mechanism.

2. Softwire46-Priority Attribute (Section 3.2):

   Depending on the deployment scenario, a client may support several different Softwire46 mechanisms. Therefore, a client may request configuration for more than one Softwire46 mechanism at a time. The Softwire46-Priority Attribute contains information allowing the client to prioritize which mechanism to use, corresponding to OPTION_S46_PRIORITY defined in [RFC8026].

3. Softwire46-Multicast Attribute (Section 3.3):

   This attribute conveys the IPv6 prefixes to be used in [RFC8114] to synthesize IPv4-embedded IPv6 addresses. The BNG uses the IPv6 prefixes returned in the RADIUS Softwire46-Multicast Attribute to populate the DHCPv6 PREFIX64 Option [RFC8115].

All of these attributes are allocated from the RADIUS "Extended Type" code space per [RFC6929].

All of these attribute designs follow [RFC6158] and [RFC6929].

This document adheres to [RFC8044] for defining the new RADIUS attributes.

3.1. Softwire46-Configuration Attribute

This attribute is of type "tlv", as defined in the RADIUS Protocol Extensions [RFC6929]. It contains some sub-attributes, with the following requirements:

The Softwire46-Configuration Attribute MUST contain one or more of the following attributes: Softwire46-MAP-E, Softwire46-MAP-T, and/or Softwire46-Lightweight-4over6.

The Softwire46-Configuration Attribute conveys the configuration information for MAP-E, MAP-T, or Lightweight 4over6. The BNG SHALL use the configuration information returned in the RADIUS attribute to populate the DHCPv6 Softwire46 Container Option(s) defined in Section 5 of [RFC7598].

The Softwire46-Configuration Attribute MAY appear in an Access-Accept packet. It MAY also appear in an Access-Request packet to indicate a preferred Softwire46 configuration. However, the server is not required to honor such a preference.

The Softwire46-Configuration Attribute MAY appear in a CoA-Request packet.

The Softwire46-Configuration Attribute MAY appear in an Accounting-Request packet.

The Softwire46-Configuration Attribute MUST NOT appear in any other RADIUS packet.

The Softwire46-Configuration Attribute is structured as follows:

Type
241 (To be confirmed by IANA).

Length
Indicates the total length, in bytes, of all fields of this attribute, including the Type, Length, Extended-Type, and the entire length of the embedded attributes.

Extended-Type
TBD1

Value
Contains one or more of the following attributes. Each attribute type may appear at most once:

Softwire46-MAP-E
For configuring MAP-E clients. For the construction of this attribute, refer to Section 3.1.1.1.

Softwire46-MAP-T
For configuring MAP-T clients. For the construction of this attribute, refer to Section 3.1.1.2.

Softwire46-Lightweight-4over6
For configuring Lightweight 4over6 clients. For the construction of this attribute, refer to Section 3.1.1.3.

The Softwire46-Configuration Attribute is associated with the following identifier: 241.Extended-Type(TBD1).

3.1.1. Softwire46 Attributes

The Softwire46 attributes can only be encapsulated in the Softwire46-Configuration Attribute. Depending on the deployment scenario, a client might request for more than one transition mechanism at a time. There MUST be at least one Softwire46 attribute encapsulated in one Softwire46-Configuration Attribute. There MUST be at most one instance of each type of Softwire46 attribute encapsulated in one Softwire46-Configuration Attribute.

There are three types of Softwire46 attributes, namely:

1. Softwire46-MAP-E (Section 3.1.1.1)

2. Softwire46-MAP-T (Section 3.1.1.2)

3. Softwire46-Lightweight 4over6 (Section 3.1.1.3)

Each type of Softwire46 attribute contains a number of sub-attributes, defined in Section 3.1.3. The hierarchy of the Softwire46 attributes is shown in Figure 1. Section 3.1.2 describes which sub-attributes are mandatory, optional, or not permitted for each defined Softwire46 attribute.

```
                                                 /1.Rule-IPv6-Prefix
 S                      /                       |
 o  /                  | 1.Softwire46-Rule -----+ 2.Rule-IPv4-Prefix
 f | Softwire46-MAP-E--+                        |
 t |                   | 2.Softwire46-BR        | 3.EA Length
 w |                   |                         \
 i |                   |                               /1.PSID-Offset
 r |                   |                              |
 e |                   | 3.Softwire46-PORTPARAMS -----+ 2.PSID-Len
 - |                    \                             |
 C |                                                  | 3.PSID
 o |                                                   \
 n |
 f |                                             /1.Rule-IPv6-Prefix
 i |                    /                       |
 g |                   | 1.Softwire46-Rule------+ 2.Rule-IPv4-Prefix
 u | Softwire46-MAP-T--+                        |
 r |                   | 2.Softwire46-DMR       | 3.EA Length
 a |                   |                         \
 t |                   |                               /1.PSID-Offset
 i |                   |                              |
 o |                   | 3.Softwire46-PORTPARAMS------+ 2.PSID-Len
 n |                    \                             |
   |                                                  | 3.PSID
 A |                                                   \
 t |
 t |                                                 /1.IPv4-Address
 r |                    /                           |
 i |                   | 1.Softwire46-V4V6Bind -----+ 2.Bind-IPv6-Prefix
 b | Softwire46-       |                             \
 u | Lightweight-4over6+ 2.Softwire46-BR               /1.PSID-Offset
 t  \                  |                              |
 e                     | 3.Softwire46-PORTPARAMS  ----+ 2.PSID-Len
                        \                             |
                                                      | 3.PSID
                                                       \
```

Figure 1: Softwire46 Attributes Hierarchy

3.1.1.1. Softwire46-MAP-E Attribute

Softwire46-MAP-E attribute is designed for carrying the configuration information for MAP-E. The structure of Softwire46-MAP-E is shown below:

TLV-Type
  1

TLV-Length
  Indicates the length of this attribute, including the TLV-Type, TLV-Length, and TLV-Value fields.

TLV-Value
  Contains a set of sub-attributes, with the following requirements:

  It MUST contain Softwire46-Rule, defined in Section 3.1.3.1.

  It MUST contain Softwire46-BR, defined in Section 3.1.3.2.

  It MAY contain Softwire46-PORTPARAMS, defined in Section 3.1.3.5.

3.1.1.2. Softwire46-MAP-T Attribute

Softwire46-MAP-T attribute is designed for carrying the configuration information for MAP-T. The structure of Softwire46-MAP-T is shown below:

TLV-Type
  2

TLV-Length
  Indicates the length of this attribute, including the TLV-Type, TLV-Length, and TLV-Value fields.

TLV-Value
  Contains a set of sub-attributes, with the following requirements:

  It MUST contain Softwire46-Rule, defined in Section 3.1.3.1.

  It MUST contain Softwire46-DMR, defined in Section 3.1.3.3.

  It MAY contain Softwire46-PORTPARAMS, defined in Section 3.1.3.5.

3.1.1.3. Softwire46-Lightweight-4over6 Attribute

Softwire46-Lightweight-4over6 attribute is designed for carrying the configuration information for Lightweight 4over6. The structure of Softwire46-Lightweight-4over6 is shown below:

TLV-Type
3

TLV-Length
Indicates the length of this attribute, including the TLV-Type, TLV-Length, and TLV-Value fields.

TLV-Value
Contains a set of sub-attributes as follows:

It MUST contain Softwire46-BR, defined in Section 3.1.3.2.

It MUST contain Softwire46-V4V6Bind, defined in Section 3.1.3.4.

It MAY contain Softwire46-PORTPARAMS, defined in Section 3.1.3.5.

3.1.2. Softwire46 Sub-Attributes

Table 2 shows which encapsulated sub-attributes are mandatory, optional, or not permitted for each defined Softwire46 attribute.

Sub-Attributes	MAP-E	MAP-T	Lightweight 4over6
Softwire46-BR	1+	0	1+
Softwire46-Rule	1	1	0
Softwire46-DMR	0	1	0
Softwire46-V4V6Bind	0	0	1
Softwire46-PORTPARAMS	0-1	0-1	0-1

Table 2: Softwire46 Sub-Attributes

The following table defines the meaning of Table 2 entries.

0	Not Permitted
0-1	Optional, zero or one instance of the attribute may be present.
1	Mandatory, only one instance of the attribute must be present.
1+	Mandatory, one or more instances of the attribute may be present.

3.1.3. Specification of the Softwire46 Sub-Attributes

3.1.3.1. Softwire46-Rule Attribute

Softwire46-Rule can only be encapsulated in Softwire46-MAP-E (Section 3.1.1.1) or Softwire46-MAP-T (Section 3.1.1.2). Depending on the deployment scenario, one Basic Mapping Rule (BMR) and zero or more Forwarding Mapping Rules (FMRs) MUST be included in one Softwire46-MAP-E or Softwire46-MAP-T.

Each type of Softwire46-Rule also contains a number of sub-attributes, including Rule-IPv6-Prefix, Rule-IPv4-Prefix, and EA-Length. The structure of the sub-attributes for Softwire46-Rule is defined in Section 3.1.4.

Defining multiple TLV-types achieves the same design goals as the "Softwire46 Rule Flags" defined in Section 4.1 of [RFC7598]. Using TLV-type set to 5 is equivalent to setting the F-flag in the OPTION_S46_RULE S46 Rule Flags field.

TLV-Type
4 Basic Mapping Rule only (not to be used for forwarding)
5 Forwarding Permitted Mapping Rule

TLV-Length
Indicates the length of this attribute, including the TLV-Type, TLV-Length, and TLV-Value fields.

Data Type
The attribute Softwire46-Rule is of type tlv (Section 3.13 of [RFC8044]).

TLV-Value
This field contains a set of attributes as follows:

Rule-IPv6-Prefix
This attribute contains the IPv6 prefix for use in the MAP rule. Refer to Section 3.1.4.1.

Rule-IPv4-Prefix
This attribute contains the IPv4 prefix for use in the MAP rule. Refer to Section 3.1.4.2.

EA-Length
This attribute contains the Embedded-Address (EA) bit length. Refer to Section 3.1.4.3.

3.1.3.2. Softwire46-BR Attribute

Softwire46-BR can only be encapsulated in Softwire46-MAP-E (Section 3.1.1.1) or Softwire46-Lightweight-4over6 (Section 3.1.1.3).

There MUST be at least one Softwire46-BR included in each Softwire46-MAP-E or Softwire46-Lightweight-4over6.

The structure of Softwire46-BR is shown below:

TLV-Type
6

TLV-Length
18 octets

Data Type
The attribute Softwire46-BR is of type ip6addr (Section 3.9 of [RFC8044]).

TLV-Value
br-ipv6-address. A fixed-length field of 16 octets that specifies the IPv6 address for the Softwire46 Border Relay (BR).

3.1.3.3. Softwire46-DMR Attribute

Softwire46-DMR may only appear in Softwire46-MAP-T (Section 3.1.1.2). There MUST be exactly one Softwire46-DMR included in one Softwire46-MAP-T.

The structure of Softwire46-DMR is shown below:

TLV-Type
7

TLV-Length
4 + length of dmr-ipv6-prefix specified in octets.

Data Type
The attribute Softwire46-DMR is of type ipv6pref (Section 3.10 of [RFC8044]).

TLV-Value
A variable-length (dmr-prefix6-len) field specifying the IPv6 prefix (dmr-ipv6-prefix) for the BR. This field is right-padded with zeros to the nearest octet boundary when dmr-prefix6-len is not divisible by 8. Prefixes with length from 0 to 96 are allowed.

3.1.3.4. Softwire46-V4V6Bind Attribute

Softwire46-V4V6Bind may only be encapsulated in Softwire46-Lightweight-4over6 (Section 3.1.1.3). There MUST be exactly one Softwire46-V4V6Bind included in each Softwire46-Lightweight-4over6.

The structure of Softwire46-V4V6Bind is shown below:

TLV-Type
8

TLV-Length
Indicates the length of this attribute, including the TLV-Type, TLV-Length, and TLV-Value fields.

Data Type
The attribute Softwire46-V4V6Bind is of type tlv (Section 3.13 of [RFC8044]).

TLV-Value
This field contains a set of attributes as follows:

IPv4-Address
This attribute contains an IPv4 address, used to specify the full or shared IPv4 address of the CE. Refer to Section 3.1.5.1.

Bind-IPv6-Prefix
This attribute contains an IPv6 prefix used to indicate which configured prefix the Softwire46 CE should use for constructing the softwire. Refer to Section 3.1.5.2.

3.1.3.5. Softwire46-PORTPARAMS Attribute

Softwire46-PORTPARAMS is optional. It is used to specify port set information for IPv4 address sharing between clients. Softwire46-PORTPARAMS MAY be included in any of the Softwire46 attributes.

The structure of Softwire46-PORTPARAMS is shown below:

TLV-Type
9

TLV-Length
Indicates the length of this attribute, including the TLV-Type, TLV-Length, and TLV-Value fields.

Data Type
The attribute Softwire46-PORTPARAMS is of type tlv (Section 3.13 of [RFC8044]).

TLV-Value
This field contains a set of attributes as follows:

PSID-Offset
This attribute specifies the numeric value for the Softwire46 algorithm’s excluded port range/offset bits (a bits). Refer to Section 3.1.6.1.

PSID-Len
This attribute specifies the number of significant bits in the PSID field (also known as ’k’). Refer to Section 3.1.6.2.

PSID
This attribute specifies PSID value. Refer to Section 3.1.6.3.

3.1.4. Sub-Attributes for Sofwtire46-Rule

There are two types of Softwire46-Rule: the Basic Mapping Rule and the Forwarding Mapping Rule, indicated by the value in the TLV-Type field of Softwire46-Rule (Section 3.1.3.1).

Each type of Softwire46-Rule also contains a number of Sub-attributes as detailed in the following sub-sections.

3.1.4.1. Rule-IPv6-Prefix Attribute

Rule-IPv6-Prefix is REQUIRED for every Softwire46-Rule. There MUST be exactly one Rule-IPv6-Prefix encapsulated in each type of Softwire46-Rule.

Rule-IPv6-Prefix follows the framed IPv6 prefix designed in [RFC3162] and [RFC8044].

The structure of Rule-IPv6-Prefix is shown below:

TLV-Type
10

TLV-Length
4 + length of rule-ipv6-prefix specified in octets.

Data Type
The attribute Rule-IPv6-Prefix is of type ipv6pref (Section 3.10 of [RFC8044]).

TLV-Value
A variable-length field that specifies an IPv6 prefix (rule-ipv6-prefix) appearing in the MAP rule.

3.1.4.2. Rule-IPv4-Prefix Attribute

This attribute is used to convey the MAP Rule IPv4 prefix. The structure of Rule-IPv4-Prefix is shown below:

TLV-Type
11

TLV-Length
4 + length of rule-ipv4-prefix specified in octets.

Data Type
The attribute Rule-IPv4-Prefix is of type ipv4pref (Section 3.11 of [RFC8044]).

TLV-Value
A variable-length field that specifies an IPv4 prefix (rule-ipv4-prefix) appearing in the MAP rule.

3.1.4.3. EA-Length Attribute

This attribute is used to convey the Embedded-Address (EA) bit length. The structure of EA-Length is shown below:

TLV-Type
12

TLV-Length
6 octets

Data Type
The attribute EA-Length is of type integer (Section 3.1 of [RFC8044]).

TLV-Value
EA-len; 32-bits long. Specifies the Embedded-Address (EA) bit length. Allowed values range from 0 to 48.

3.1.5. Attributes for Softwire46-v4v6Bind

3.1.5.1. IPv4-Address Attribute

The IPv4-Address MAY be used to specify the full or shared IPv4 address of the CE.

The structure of IPv4-Address is shown below:

TLV-Type
13

TLV-Length
6 octets

Data Type
The attribute IPv4-Address is of type ipv4addr (Section 3.8 of [RFC8044]).

TLV-Value
32-bits long. Specifies the IPv4 address (ipv4-address) to appear in Softwire46-V4V6Bind (Section 3.1.3.4).

3.1.5.2. Bind-IPv6-Prefix Attribute

The Bind-IPv6-Prefix is used by the CE to identify the correct IPv6 prefix to be used as the tunnel source.

The structure of Bind-IPv6-Prefix is shown below:

TLV-Type
14

TLV-Length
4 + length of bind-ipv6-prefix specified in octets.

Data Type
The attribute Bind-IPv6-Prefix is of type ipv6pref (Section 3.10 of [RFC8044]).

TLV-Value
A variable-length field specifying the IPv6 prefix or address for the Softwire46 CE (bind-ipv6-prefix). This field is right-padded with zeros to the nearest octet boundary when the prefix length is not divisible by 8.

3.1.6. Attributes for Softwire46-PORTPARAMS

3.1.6.1. PSID-Offset Attribute

This attribute is used to convey the Port Set Identifier offset as defined in [RFC7597]. This attribute is encoded in 32 bits as per the recommendation in Appendix A.2.1 of [RFC6158].

The structure of PSID-Offset is shown below:

TLV-Type
15

TLV-Length
6 octets

Data Type
The attribute PSID-Offset is of type integer (Section 3.1 of [RFC8044]).

TLV-Value
Contains the PSID-Offset (8-bits) right justified, and the unused bits in this field MUST be set to zero. This field specifies the numeric value for the Softwire46 algorithm's excluded port range/offset bits (a bits), as per Section 5.1 of [RFC7597].
Default values for this field are specific to the Softwire mechanism being implemented and are defined in the relevant specification document.

3.1.6.2. PSID-Len Attribute

This attribute is used to convey the PSID length as defined in [RFC7597]. This attribute is encoded in 32 bits as per the recommendation in Appendix A.2.1 of [RFC6158].

The structure of PSID-Len is shown below:

TLV-Type
16

TLV-Length
6 octets

Data Type
The attribute PSID-Len is of type integer (Section 3.1 of [RFC8044]).

TLV-Value
Contains the PSID-len (8-bits) right justified, and the unused bits in this field MUST be set to zero. This field specifies the number of significant bits in the PSID field (also known as 'k'). When set to 0, the PSID field is to be ignored. After the first 'a' bits, there are k bits in the port number representing the value of the PSID. Subsequently, the address sharing ratio would be 2^k.

3.1.6.3. PSID Attribute

This attribute is used to convey the PSID as defined in [RFC7597]. This attribute is encoded in 32 bits as per the recommendation in Appendix A.2.1 of [RFC6158].

The structure of PSID is shown below:

TLV-Type
17

TLV-Length
6 octets

Data Type
The attribute PSID is of type integer (Section 3.1 of [RFC8044]).

TLV-Value
Contains the PSID (16-bits) right justified, and the unused bits in this field MUST be set to zero.
The PSID value algorithmically identifies a set of ports assigned to a CE. The first k bits on the left of this 2-octet field is the PSID value. The remaining (16-k) bits on the right are padding zeros.

3.2. Softwire46-Priority Attribute

The Softwire46-Priority Attribute includes an ordered list of Softwire46 mechanisms allowing the client to prioritize which mechanism to use, corresponding to OPTION_S46_PRIORITY defined in [RFC8026]. The following requirements apply:

The Softwire46-Priority Attribute MAY appear in an Access-Accept packet. It MAY also appear in an Access-Request packet.

The Softwire46-Priority Attribute MAY appear in a CoA-Request packet.

The Softwire46-Priority Attribute MAY appear in an Accounting-Request packet.

The Softwire46-Priority Attribute MUST NOT appear in any other RADIUS packet.

The Softwrie46-Priority Attribute is structured as follows:

Type
241 (To be confirmed by IANA)

Length
Indicates the length of this attribute, including the Type, Length, Extended-Type and Value fields.

Extended-Type
TBD5

TLV-Value
The attribute includes one or more Softwire46-Option-Code TLVs:
A Softwire46-Priority Attribute MUST contain at least one Softwire46-Option-Code TLV (Section 3.2.1).

Softwire46 mechanisms are prioritized in the appearance order of the in the Softwire46-Priority Attribute. That is, the first-appearing mechanism is most preferred.

The Softwire46-Priority Attribute is associated with the following identifier: 241.Extended-Type (TBD5).

3.2.1. Softwire46-Option-Code

This attribute is used to convey an option code assigned to a Softwire46 mechanism [RFC8026]. This attribute is encoded in 32 bits as per the recommendation in Appendix A.2.1 of [RFC6158].

The structure of Softwire46-Option-Code is shown below:

TLV-Type
18

TLV-Length
6 octets

Data Type
The attribute Softwire46-Option-Code is of type integer (Section 3.1 of [RFC8044]).

TLV-Value
A 32-bit IANA-registered option code representing a Softwire46 mechanism (Softwire46-option-code). The codes and their corresponding Softwire46 mechanisms are listed in Section 7.3.

3.3. Softwire46-Multicast Attribute

The Softwire46-Multicast Attribute conveys the IPv6 prefixes to be used to synthesize multicast and unicast IPv4-embedded IPv6 addresses as per [RFC8114]. This attribute is of type "tlv" and contains additional TLVs. The following requirements apply:

The BNG SHALL use the IPv6 prefixes returned in the RADIUS Softwire46-Multicast Attribute to populate the DHCPv6 PREFIX64 Option [RFC8115].

This attribute MAY be used in Access-Request packets as a hint to the RADIUS server. For example, if the BNG is pre-configured for Softwire46-Multicast, these prefixes may be inserted in the attribute. The RADIUS server MAY ignore the hint sent by the BNG, and it MAY assign a different Softwire46-Multicast Attribute.

The Softwire46-Multicast Attribute MAY appear in an Access-Request, Access-Accept, CoA-Request, and Accounting-Request packet.

The Softwire46-Multicast Attribute MUST NOT appear in any other RADIUS packet.

The Softwire46-Multicast Attribute MAY contain ASM-Prefix64 (Section 3.3.1), SSM-Prefix64 (Section 3.3.2), and U-Prefix64 (Section 3.3.3).

The Softwire46-Multicast Attribute MUST include ASM-Prefix64 or SSM-Prefix64, and it MAY include both.

The U-Prefix64 MUST be present when SSM-Prefix64 is present. U-Prefix64 MAY be present when ASM-Prefix64 is present.

The Softwire46-Multicast Attribute is structured as follows:

Type
241 (To be confirmed by IANA)

Length
This field indicates the total length in bytes of all fields of this attribute, including the Type, Length, Extended-Type, and the entire length of the embedded attributes.

Extended-Type
TBD6

Value
This field contains a set of attributes as follows:

ASM-Prefix64
This attribute contains the Any-Source Multicast (ASM) IPv6 prefix. Refer to Section 3.3.1.

SSM-Prefix64
This attribute contains the Source-Source Multicast (SSM) IPv6 prefix. Refer to Section 3.3.2.

U-Prefix64
This attribute contains the IPv4 prefix used for address translation. Refer to Section 3.3.3.

The Softwire46-Multicast Attribute is associated with the following identifier: 241.Extended-Type(TBD6).

3.3.1. ASM-Prefix64 Attribute

The ASM-Prefix64 attribute is structured as follows:

TLV-Type
19

TLV-Length
16 octets. The length of asm-prefix64 must be /96 [RFC8115].

Data Type
The attribute ASM-Prefix64 is of type ipv6prefix (Section 3.10 of [RFC8044]).

TLV-Value
This field specifies the IPv6 multicast prefix (asm-prefix64) to be used to synthesize the IPv4-embedded IPv6 addresses of the multicast groups in the ASM mode. The conveyed multicast IPv6 prefix MUST belong to the ASM range.

3.3.2. SSM-Prefix64 Attribute

The SSM-Prefix64 attribute is structured as follows:

Type
  20

TLV-Length
  16 octets. The length of ssm-prefix64 must be /96 [RFC8115].

Data Type
  The attribute SSM-Prefix64 is of type ipv6prefix (Section 3.10 of [RFC8044]).

TLV-Type
  This field specifies the IPv6 multicast prefix (ssm-prefix64) to be used to synthesize the IPv4-embedded IPv6 addresses of the multicast groups in the SSM mode. The conveyed multicast IPv6 prefix MUST belong to the SSM range.

3.3.3. U-Prefix64 Attribute

The structure of U-Prefix64 is shown below:

TLV-Type
  21

TLV-Length
  4 + length of unicast-prefix. As specified in [RFC6052], the unicast-prefix prefix-length MUST be set to 32, 40, 48, 56, 64, or 96.

Data Type
  The attribute U-Prefix64 is of type ipv6prefix (Section 3.10 of [RFC8044]).

TLV-Value
  This field identifies the IPv6 unicast prefix (u-prefix64) to be used in SSM mode for constructing the IPv4-embedded IPv6 addresses representing the IPv4 multicast sources in the IPv6 domain. It may also be used to extract the IPv4 address from the received multicast data flows.

4. A Sample Configuration Process with RADIUS

Figure 2 illustrates how the RADIUS and DHCPv6 protocols interwork to provide CE with softwire configuration information.

```
   CE                             BNG                           AAA Server
    |                              |                                |
    |-------1.DHCPv6 Solicit------->|                                |
    |(ORO with unicast and/or m'cast|                                |
    |    container option code(s)) |                                |
    |                              |                                |
    |                              |-------2.Access-Request------->|
    |                              |   (Softwire46-Configuration    |
    |                              |       Attribute and/or         |
    |                              |Softwire46-Multicast Attribute) |
    |                              |                                |
    |                              |<------3.Access-Accept---------|
    |                              |   (Softwire46-Configuration    |
    |                              |       Attribute and/or         |
    |                              |Softwire46-Multicast Attribute) |
    |                              |                                |
    |<----4.DHCPv6 Advertisement----|                                |
    |      (container option(s))   |                                |
    |                              |                                |
    |-------5.DHCPv6  Request------>|                                |
    |      (container Option(s))   |                                |
    |                              |                                |
    |<--------6.DHCPv6 Reply--------|                                |
    |      (container option(s))   |                                |
    |                              |                                |
                DHCPv6                          RADIUS
```

Figure 2: Interaction between DHCPv6 and AAA Server with RADIUS authentication

1. The CE creates a DHCPv6 Solicit message. For unicast softwire configuration, the message includes an OPTION_REQUEST_OPTION (6) with the Softwire46 Container option code(s) as defined in [RFC7598]. OPTION_S46_CONT_MAPE (94) should be included for MAP-E, OPTION_S46_CONT_MAPT (95) for MAP-T, and OPTION_S46_CONT_LW (96) for Lightweight 4over6. For multicast configuration, the option number for OPTION_V6_PREFIX64 (113) is included in the client's ORO. The message is sent to the BNG.

2. On receipt of the Solicit message, the BNG constructs a RADIUS Access-Request message containing a User-Name Attribute (1) (containing either a CE MAC address, interface-id, or both), a User-Password Attribute (2) (with a pre-configured shared password between the CE and AAA server as defined in [RFC2865]). The Softwire46-Configuration Attribute and/or Softwire46-Multicast Attribute are also included (as requested by the client). The resulting message is sent to the AAA server.

3. The AAA server authenticates the request. If this is successful, and suitable configuration is available, an Access-Accept message is sent to the BNG containing the requested Softwire46-Configuration Attribute or Softwire46-Multicast Attribute. It is the responsibility of the AAA server to ensure the consistency of the provided configuration.

4. The BNG maps the received softwire configuration into the corresponding fields in the DHCPv6 softwire configuration option(s). These are included in the DHCPv6 Advertise message which is sent to the CE.

5. The CE sends a DHCPv6 Request message. In the ORO, the option code(s) of any of the required softwire options that were received in the Advertise message are included.

6. The BNG sends a DHCPv6 Reply message to the client containing the softwire container option(s) enumerated in the ORO.

The authorization operation could be done independently, after the authentication process. In this case, steps 1-5 are completed as above, then the following steps are performed:

6a. When the BNG receives the DHCPv6 Request, it constructs a RADIUS Access-Request message, which contains a Service-Type Attribute (6) with the value "Authorize Only" (17), the corresponding Softwire46-Configuration Attribute, and a State Attribute obtained from the previous authentication process according to [RFC5080]. The resulting message is sent to the AAA server.

7a. The AAA checks the authorization request. If it is approved, an Access-Accept message is returned to the BNG with the corresponding Softwire46-Configuration Attribute.

8a. The BNG sends a Reply message to the client containing the softwire container options enumerated in the ORO.

In addition to the above, the following points need to be considered:

- In the configuration message flows described above the Message-Authenticator (type 80) [RFC2869] should be used to protect both Access-Request and Access-Accept messages.

- If the BNG does not receive the corresponding Softwire46-Configuration Attribute in the Access-Accept message it may fall back to creating the DHCPv6 softwire configuration options using pre-configured Softwire46 configuration, if this is present.

- o If the BNG receives an Access-Reject from the AAA server, then Softwire46 configuration must not be supplied to the client.

- o As specified in [RFC8415], Section 18.2.5, "Creation and Transmission of Rebind Messages", if the DHCPv6 server to which the DHCPv6 Renew message was sent at time T1 has not responded by time T2, the CE (DHCPv6 client) should enter the Rebind state and attempt to contact any available server. In this situation, a secondary BNG receiving the DHCPv6 message must initiate a new Access-Request message towards the AAA server. The secondary BNG includes the Softwire46-Configuration Attribute in this Access-Request message.

- o For Lightweight 4over6, the CE's binding state needs to be synchronized between the clients and the Lightweight AFTR (lwAFTR)/BR. This can be achieved in two ways: static pre-configuration of the bindings on both the AAA server and lwAFTR, or on-demand whereby the AAA server updates the lwAFTR with the CE's binding state as it is created or deleted.

In some deployments, the DHCP server may use the Accounting-Request to report to a AAA server the softwire configuration returned to a requesting host. It is the responsibility of the DHCP server to ensure the consistency of the configuration provided to requesting hosts. Reported data to a AAA server may be required for various operational purposes (e.g., regulatory).

A configuration change (e.g., BR address) may result in an exchange of CoA-Requests between the BNG and the AAA server as shown in Figure 3. Concretely, when the BNG receives a CoA-Request message containing Softwire46 attributes, it sends a DHCPv6 Reconfigure message to the appropriate CE to inform that CE that an updated configuration is available. Upon receipt of such message, the CE sends a DHCPv6 Renew or Information-Request in order to receive the updated Softwire46 configuration. In deployments where the BNG embeds a DHCPv6 relay, CoA-Requests can be used following the procedure specified in [RFC6977].

```
        CE                            BNG                       AAA Server
         |                             |                             |
         |---DHCPv6 Solicit--------->|                             |
         |                             |---Access-Request---------->|
         |                             |<--Access-Accept------------|
         |                             |(Softwire46-Configuration    |
         |                             |    Attribute ...)           |
                                     ....
         |                             |                             |
         |                             |<-----CoA-Request-----------|
         |                             |(Softwire46-Configuration    |
         |                             |    Attribute ...)           |
         |                             |------CoA-Response--------->|
         |<--DHCPv6 Reconfigure------|                             |
         |                             |                             |
                                     ....
```

Figure 3: Change of Configuration Example

5. Table of Attributes

This document specifies three new RADIUS attributes, and their formats are as follows:

- o Softwire46-Configuration Attribute: 241.TBD1
- o Softwire46-Priority Attribute: 241.TBD5
- o Softwire46-Multicast Attribute: 241.TBD6

Table 3 describes which attributes may be found, in which kinds of packets and in what quantity.

Request	Accept	Reject	Challenge	Acct Req	CoA-Req	#	Attribute
0-1	0-1	0	0	0-1	0-1	241.TBD1	Softwire46-Configuration
0-1	0-1	0	0	0-1	0-1	241.TBD5	Softwire46-Priority
0-1	0-1	0	0	0-1	0-1	241.TBD6	Softwire46-Multicast

Table 3: Table of Attributes

6. Security Considerations

Section 9 of [RFC7596] discusses security issues related to Lightweight 4over6, Section 10 of [RFC7597] discusses security issues related to MAP-E, Section 13 of [RFC7599] discusses security issues related to MAP-T, and Section 9 of [RFC8114] discusses security issues related to the delivery of IPv4 multicast services to IPv4 clients over an IPv6 multicast network.

This document does not introduce any security issues inherently different from those already identified in Section 8 of [RFC2865] and Section 6 of [RFC5176] for CoA messages. Known security vulnerabilities of the RADIUS protocol discussed in Section 7 of [RFC2607] and Section 7 of [RFC2869] apply to this specification. These well-established properties of the RADIUS protocol place some limitations on how it can safely be used, since there is some inherent requirement to trust the counterparty to not misbehave.

Accordingly, this document targets deployments where a trusted relationship is in place between the RADIUS client and server with communication optionally secured by IPsec or Transport Layer Security (TLS) [RFC6614]. The use of IPsec [RFC4301] for providing security when RADIUS is carried in IPv6 is discussed in [RFC3162].

Security considerations for interactions between a Softwire46 CE and the BNG are discussed in Section 9 of [RFC7598] (DHCPv6 options for configuration of softwire46 address and port-mapped clients), Section 3 of [RFC8026] (DHCPv6-based Softwire46 prioritization mechanism), and Section 5 of [RFC8115] (DHCPv6 options for configuration of IPv4-embedded IPv6 prefixes).

7. IANA Considerations

IANA is requested to make new code point assignments for RADIUS attributes as described in the following subsections. The assignments should use the RADIUS registry available at https://www.iana.org/assignments/radius-types/.

7.1. New RADIUS Attributes

This document requests IANA to assign the Attribute Types defined in this document from the RADIUS namespace as described in the "IANA Considerations" section of [RFC3575], in accordance with BCP 26 [RFC8126].

This document requests that IANA register three new RADIUS attributes, from the "Short Extended Space" of [RFC6929]. The attributes are: Softwire46-Configuration Attribute, Softwire46-Priority Attribute, and Softwire46-Multicast Attribute:

Type	Description	Data Type	Reference
241.TBD1	Softwire46-Configuration	tlv	Section 3.1
241.TBD5	Softwire46-Priority	tlv	Section 3.2
241.TBD6	Softwire46-Multicast	tlv	Section 3.3

7.2. RADIUS Softwire46 Configuration and Multicast Attributes

IANA is requested to create a new registry called "RADIUS Softwire46 Configuration and Multicast Attributes".

All attributes in this registry have one or more parent RADIUS attributes in nesting (refer to [RFC6929]).

This registry must be initially populated with the following values:

Value	Description	Data Type	Reference
0	Reserved		
1	Softwire46-MAP-E	tlv	Section 3.1.1.1
2	Softwire46-MAP-T	tlv	Section 3.1.1.2
3	Softwire46-Lightweight-4over6	tlv	Section 3.1.1.3
4	Softwire46-Rule (BMR)	tlv	Section 3.1.3.1
5	Softwire46-Rule (FMR)	tlv	Section 3.1.3.1
6	Softwire46-BR	ipv6addr	Section 3.1.3.2
7	Softwire46-DMR	ipv6prefix	Section 3.1.3.3
8	Softwire46-V4V6Bind	tlv	Section 3.1.3.4
9	Softwire46-PORTPARAMS	tlv	Section 3.1.3.5
10	Rule-IPv6-Prefix	ipv6prefix	Section 3.1.4.1
11	Rule-IPv4-Prefix	ipv4prefix	Section 3.1.4.2
12	EA-Length	integer	Section 3.1.4.3
13	IPv4-Address	ipv4addr	Section 3.1.5.1
14	Bind-IPv6-Prefix	ipv6prefix	Section 3.1.5.2
15	PSID-Offset	integer	Section 3.1.6.1
16	PSID-Len	integer	Section 3.1.6.2
17	PSID	integer	Section 3.1.6.3
18	Softwire46-Option-Code	integer	Section 3.2.1
19	ASM-Prefix64	ipv6prefix	Section 3.3.1
20	SSM-Prefix64	ipv6prefix	Section 3.3.2
21	U-Prefix64	ipv6prefix	Section 3.3.3
22-255	Unassigned		

The registration procedure for this registry is Standards Action as defined in [RFC8126].

7.3. Softwire46 Mechanisms and Their Identifying Option Codes

The Softwire46-Priority Attribute conveys an ordered list of option codes assigned to Softwire46 mechanisms, for which IANA is requested to create and maintain a new registry entitled "Option Codes Permitted in the Softwire46-Priority Attribute".

Table 4 shows the initial version of allowed option codes, and the Softwire46 mechanisms that they represent. The option code for DS-Lite is derived from the IANA allocated RADIUS Attribute Type value for DS-Lite [RFC6519]. The option codes for MAP-E, MAP-T, and Lightweight 4over6 are the TLV-Type values for the MAP-E, MAP-T, and Lightweight 4over6 attributes defined in Section 3.1.1.

Option Code	Softwire46 Mechanism	Reference
1	MAP-E	RFC7597
2	MAP-T	RFC7599
3	Lightweight 4over6	RFC7596
144	DS-Lite	RFC6519

Table 4: Option Codes to S46 Mechanisms

Additional option codes may be added to this list in the future using the IETF Review process described in Section 4.8 of [RFC8126].

8. Contributing Authors

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

Email: leo.liubing@huawei.com

Peter Deacon
IEA Software, Inc.
P.O. Box 1170
Veradale, WA 99037
USA

Email: peterd@iea-software.com

Qiong Sun
China Telecom

Beijing China

Email: sunqiong@ctbri.com.cn

Qi Sun
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China
Phone: +86-10-6278-5822

Email: sunqibupt@gmail.com

Cathy Zhou
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129

Email: cathy.zhou@huawei.com

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

Email: Tina.Tsou.Zouting@huawei.com

ZiLong Liu
Tsinghua University
Beijing 100084
P.R.China
Phone: +86-10-6278-5822

Email: liuzilong8266@126.com

Yong Cui
Tsinghua University
Beijing 100084
P.R.China
Phone: +86-10-62603059

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

9. Acknowledgements

The authors would like to thank the valuable comments made by Peter Lothberg, Wojciech Dec, Ian Farrer, Suresh Krishnan, Qian Wang, Wei Meng, Cui Wang, Alan Dekok, Stefan Winter, and Yu Tianpeng to this document.

This document was merged with [I-D.sun-softwire-lw4over6-radext] and [I-D.wang-radext-multicast-radius-ext], thanks to everyone who contributed to this document.

This document was produced using the xml2rfc tool [RFC7991].

Many thanks to Al Morton, Bernie Volz, Joel Halpern, and Donald Eastlake for the review.

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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <https://www.rfc-editor.org/info/rfc2865>.

[RFC3162] Aboba, B., Zorn, G., and D. Mitton, "RADIUS and IPv6", RFC 3162, DOI 10.17487/RFC3162, August 2001, <https://www.rfc-editor.org/info/rfc3162>.

[RFC3575] Aboba, B., "IANA Considerations for RADIUS (Remote Authentication Dial In User Service)", RFC 3575, DOI 10.17487/RFC3575, July 2003, <https://www.rfc-editor.org/info/rfc3575>.

[RFC5080] Nelson, D. and A. DeKok, "Common Remote Authentication Dial In User Service (RADIUS) Implementation Issues and Suggested Fixes", RFC 5080, DOI 10.17487/RFC5080, December 2007, <https://www.rfc-editor.org/info/rfc5080>.

[RFC5176] Chiba, M., Dommety, G., Eklund, M., Mitton, D., and B. Aboba, "Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)", RFC 5176, DOI 10.17487/RFC5176, January 2008, <https://www.rfc-editor.org/info/rfc5176>.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, DOI 10.17487/RFC6052, October 2010, <https://www.rfc-editor.org/info/rfc6052>.

[RFC6158] DeKok, A., Ed. and G. Weber, "RADIUS Design Guidelines", BCP 158, RFC 6158, DOI 10.17487/RFC6158, March 2011, <https://www.rfc-editor.org/info/rfc6158>.

[RFC6929] DeKok, A. and A. Lior, "Remote Authentication Dial In User Service (RADIUS) Protocol Extensions", RFC 6929, DOI 10.17487/RFC6929, April 2013, <https://www.rfc-editor.org/info/rfc6929>.

[RFC8026] Boucadair, M. and I. Farrer, "Unified IPv4-in-IPv6 Softwire Customer Premises Equipment (CPE): A DHCPv6-Based Prioritization Mechanism", RFC 8026, DOI 10.17487/RFC8026, November 2016, <https://www.rfc-editor.org/info/rfc8026>.

[RFC8044] DeKok, A., "Data Types in RADIUS", RFC 8044, DOI 10.17487/RFC8044, January 2017, <https://www.rfc-editor.org/info/rfc8044>.

[RFC8115] Boucadair, M., Qin, J., Tsou, T., and X. Deng, "DHCPv6 Option for IPv4-Embedded Multicast and Unicast IPv6 Prefixes", RFC 8115, DOI 10.17487/RFC8115, March 2017, <https://www.rfc-editor.org/info/rfc8115>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

10.2. Informative References

[I-D.sun-softwire-lw4over6-radext]
Xie, C., Sun, Q., Qiong, Q., Zhou, C., Tsou, T., and Z. Liu, "Radius Extension for Lightweight 4over6", draft-sun-softwire-lw4over6-radext-01 (work in progress), March 2014.

[I-D.wang-radext-multicast-radius-ext]
Wang, Q., Meng, W., Wang, C., and M. Boucadair, "RADIUS Extensions for IPv4-Embedded Multicast and Unicast IPv6 Prefixes", draft-wang-radext-multicast-radius-ext-00 (work in progress), December 2015.

[RFC2607] Aboba, B. and J. Vollbrecht, "Proxy Chaining and Policy Implementation in Roaming", RFC 2607, DOI 10.17487/RFC2607, June 1999, <https://www.rfc-editor.org/info/rfc2607>.

[RFC2869] Rigney, C., Willats, W., and P. Calhoun, "RADIUS Extensions", RFC 2869, DOI 10.17487/RFC2869, June 2000, <https://www.rfc-editor.org/info/rfc2869>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

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

[RFC6346] Bush, R., Ed., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, DOI 10.17487/RFC6346, August 2011, <https://www.rfc-editor.org/info/rfc6346>.

[RFC6519] Maglione, R. and A. Durand, "RADIUS Extensions for Dual-Stack Lite", RFC 6519, DOI 10.17487/RFC6519, February 2012, <https://www.rfc-editor.org/info/rfc6519>.

[RFC6614] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "Transport Layer Security (TLS) Encryption for RADIUS", RFC 6614, DOI 10.17487/RFC6614, May 2012, <https://www.rfc-editor.org/info/rfc6614>.

[RFC6977] Boucadair, M. and X. Pougnard, "Triggering DHCPv6 Reconfiguration from Relay Agents", RFC 6977, DOI 10.17487/RFC6977, July 2013, <https://www.rfc-editor.org/info/rfc6977>.

[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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc7598>.

[RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July 2015, <https://www.rfc-editor.org/info/rfc7599>.

[RFC7991] Hoffman, P., "The "xml2rfc" Version 3 Vocabulary", RFC 7991, DOI 10.17487/RFC7991, December 2016, <https://www.rfc-editor.org/info/rfc7991>.

[RFC8114] Boucadair, M., Qin, C., Jacquenet, C., Lee, Y., and Q. Wang, "Delivery of IPv4 Multicast Services to IPv4 Clients over an IPv6 Multicast Network", RFC 8114, DOI 10.17487/RFC8114, March 2017, <https://www.rfc-editor.org/info/rfc8114>.

Appendix A. DHCPv6 to RADIUS Field Mappings

The following sections detail the mappings between the softwire DHCPv6 option fields and the relevant RADIUS attributes as defined in this document.

A.1. OPTION_S46_RULE (89) to Softwire46-Rule Sub-TLV Field Mappings

OPTION_S46_RULE Field	Softwire46-Rule Name	TLV Subfield
flags	N/A	TLV-type (TBD7, TBD8)
ea-len	EA-Length	EA-len
prefix4-len	Rule-IPv4-Prefix	Prefix-Length
ipv4-prefix	Rule-IPv4-Prefix	rule-ipv4-prefix
prefix6-len	Rule-IPv6-Prefix	Prefix-Length
ipv6-prefix	Rule-IPv6-Prefix	rule-ipv6-prefix

A.2. OPTION_S46_BR (90) to Softwire46-BR Field Mappings

OPTION_S46_BR Field	Softwire46-BR Subfield
br-ipv6-address	br-ipv6-address

A.3. OPTION_S46_DMR (91) to Softwire46-DMR

OPTION_S46_BR Field	Softwire46-DMR Subfield
dmr-prefix6-len	dmr-prefix6-len
dmr-ipv6-prefix	dmr-ipv6-prefix

A.4. OPTION_S46_V4V6BIND (92) to Softwire46-V4V6Bind

OPTION_S46_V4V6BIND Field	Softwire46-V4V6Bind Name	TLV Subfield
ipv4-address	IPv4-Address	ipv4-address
bindprefix6-len	Bind-IPv6-Prefix	Prefix-Length
bind-ipv6-prefix	Bind-IPv6-Prefix	bind-ipv6-prefix

A.5. OPTION_S46_PORTPARAMS (93) to Softwire46-PORTPARAMS Field Mappings

OPTION_S46_PORTPARAMS Field	Softwire46-PORTPARAMS Name	TLV Subfield
offset PSID-len PSID	PSID-Offset PSID-Len PSID	PSID-Offset PSID-len PSID

A.6. OPTION_S46_PRIORITY (111) to Softwire46-PORTPARAMS Field Mappings

OPTION_S46_PRIORITY Field	Softwire46-Priority Attribute Subfield
s46-option-code	Softwire46-option-code

A.7. OPTION_V6_PREFIX64 (113) to Softwire46-Multicast Attribute Field Mappings

OPTION_V6_PREFIX64 Field	Softwire46-Multicast Attribute TLV Name	TLV Subfield
asm-length ASM_mPrefix64 ssm-length SSM_mPrefix64 unicast-length uPrefix64	ASM-Prefix64 ASM-Prefix64 SSM-Prefix64 SSM-Prefix64 U-Prefix64 U-Prefix64	Prefix-Length asm-prefix64 Prefix-Length ssm-prefix64 Prefix-Length u-prefix64

Authors' Addresses

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

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

Email: eleven711711@foxmail.com

Chongfeng Xie
China Telecom
Beijing
P.R. China

Email: xiechf.bri@chinatelecom.cn

Tianxiang Li
Tsinghua University
Beijing 100084
P.R.China

Email: peter416733@gmail.com

Mohamed Boucadair (editor)
Orange
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Softwire WG
Internet-Draft
Intended status: Standards Track
Expires: April 3, 2017

M. Boucadair
Orange
I. Farrer
Deutsche Telekom

September 30, 2016

Unified IPv4-in-IPv6 Softwire CPE: A DHCPv6-based Prioritization Mechanism

draft-ietf-softwire-unified-cpe-08

Abstract

In IPv6-only provider networks, transporting IPv4 packets encapsulated in IPv6 is a common solution to the problem of IPv4 service continuity. 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 specifies a DHCPv6 option whereby a single CPE can interwork with all of the standardized and proposed approaches to providing encapsulated IPv4 in IPv6 services by providing a prioritization mechanism.

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 (e.g., [RFC6333], [RFC7596], or [RFC7597]). These approaches, referred to as 'S46 mechanisms' in this document, exist in order to meet the particular deployment, scaling, addressing and other requirements of different service provider's networks.

A common feature shared between all of the differing modes is the integration of softwire tunnel end-point functionality into the

Customer Premise Equipment (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 S46 mechanism enabled in order to support a diverse CPE population with differing client functionality, such as during a migration between mechanisms, 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.

A number of DHCPv6 options for the provisioning of softwires have been standardized:

RFC6334 Defines DHCPv6 option 64 for configuring Basic Bridging BroadBand (B4, [RFC6333]) elements with the IPv6 address of the Address Family Transition Router (AFTR, [RFC6333]).

RFC7341 Defines DHCPv6 option 88 for configuring the address of a DHCPv4 over DHCPv6 server, which can then be used by a softwire client for obtaining further configuration.

RFC7598 Defines DHCPv6 options 94, 95 and 96 for provisioning Mapping of Address and Port with Encapsulation (MAP-E, [RFC7597]), Mapping of Address and Port using Translation (MAP-T, [RFC7599]), and Lightweight 4over6 [RFC7596] respectively.

This document describes a DHCPv6 based prioritization method whereby a CPE which supports several S46 mechanisms and receives configuration for more than one can prioritise which mechanism to use. The method requires no server side logic to be implemented and only uses a simple S46 mechanism prioritization to be implemented in the CPE.

The prioritization method as described here does not provide redundancy between S46 mechanisms for the client. I.e. If the highest priority S46 mechanism which has been provisioned to the client is not available for any reason, the means for identifying this and falling back to the S46 mechanism with the next highest priority is not in the scope of this document.

1.1. Terminology

This document makes use of the following terms:

- o Address Family Transition Router (AFTR): is the IPv4-in-IPv6 tunnel termination point and the NAT44 function deployed in the operator's network [RFC6333].
- o Border Relay (BR): a MAP-enabled router managed by the service provider at the edge of a MAP domain. A BR has at least an IPv6-enabled interface and an IPv4 interface connected to the native IPv4 network [RFC7597].
- o Customer Premise Equipment (CPE): denotes the equipment at the customer edge that terminates the customer end of an IPv6 transitional tunnel. In some documents (e.g., [RFC7597]), this functional entity is called CE (Customer Edge).

1.2. Rationale

The following rationale has been adopted for this document:

(1) Simplify solution migration paths: Define unified CPE behavior, allowing for smooth migration between the different s46 mechanisms.

(2) Deterministic CPE co-existence behavior: Specify the behavior when several S46 mechanisms co-exist in the CPE.

(3) Deterministic service provider co-existence behavior: Specify the behavior when several modes co-exist in the service providers network.

(4) Re-usability: Maximize the re-use of existing functional blocks including tunnel end-points, port restricted NAPT44, forwarding behavior, etc.

(5) Solution agnostic: Adopt neutral terminology and avoid (as far as possible) overloading the document with solution-specific terms.

(6) Flexibility: Allow operators to compile CPE software only for the mode(s) necessary for their chosen deployment context(s).

(7) Simplicity: Provide a model that allows operators to only implement the specific mode(s) that they require without the additional complexity of unneeded modes.

1.3. DHCPv6 S46 Priority Option

The S46 Priority Option is used to convey a priority order of IPv4 service continuity mechanisms. Figure 1 shows the format of the S46 Priority 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_S46_PRIORITY      |         option-length         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |        s46-option-code        |        s46-option-code        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              ...              |        s46-option-code        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: S46 Priority Option

- option-code: OPTION_S46_PRIORITY (TBD)
- option-length: >=2 and a multiple of 2, in octets.
- s46-option-code: 16-bits long IANA registered option code of the DHCPv6 option which is used to identify the softwire mechanism. S46 mechanism are prioritized in the appearance order in the S46 Priority Option.

Codes in OPTION_S46_PRIORITY are processed in order; in the event that a client receives more than one s46-option-code with a particular value, this should be considered as invalid. DHCP servers MAY validate the list of s46-option-code values to detect invalid values and duplicates. The option MUST contain at least one s46-option-code.

1.4. DHCPv6 Client Behavior

Clients MAY request option OPTION_S46_PRIORITY, as defined in [RFC3315], Sections 17.1.1, 18.1.1, 18.1.3, 18.1.4, 18.1.5, and 22.7. As a convenience to the reader, we mention here that the client includes requested option codes in the Option Request Option.

Upon receipt of a DHCPv6 Advertise message from the server containing OPTION_S46_PRIORITY the client performs the following steps:

1. Check the contents of the DHCPv6 message for options containing valid S46 mechanism configuration. A candidate list of possible S46 mechanisms is created from these option codes.
2. Check the contents of OPTION_S46_PRIORITY for the DHCPv6 option codes contained in the included s46-option-code fields. From this, an S46 mechanism priority list is created, ordered from highest to lowest following the appearance order.
3. Sequentially check the priority list against the candidate list until a match is found.
4. When a match is found, the client MUST configure the resulting S46 mechanism.

In the event that no match is found between the priority list and the candidate list, the client MAY proceed with configuring one or more of the provisioned S46 softwire mechanism(s). In this case, which mechanism(s) are chosen by the client is implementation-specific and not defined here.

If an invalid OPTION_S46_PRIORITY option is received, the client MAY proceed with configuring the provisioned S46 mechanisms as if OPTION_S46_PRIORITY had not been received.

If an unknown option code is received in OPTION_S46_PRIORITY option, the client MUST skip it and continue processing other listed option codes if they exist. The initial option codes that are allowed to be included in a OPTION_S46_PRIORITY option are listed in Section 4.1.

1.5. DHCPv6 Server Behavior

Sections 17.2.2 and 18.2 of [RFC3315] govern server operation in regards to option assignment. As a convenience to the reader, we mention here that the server will send a particular option code only if configured with specific values for that option code and if the client requested it.

Option OPTION_S46_PRIORITY is a singleton. Servers MUST NOT send more than one instance of the OPTION_S46_PRIORITY option.

2. Operator Deployment Considerations for Deploying Multiple Sotfwire Mechanisms

The following sub-sections describe some considerations for operators who are planning on implementing multiple softwire mechanisms in their network (e.g., during a migration between mechanisms).

2.1. Client Address Planning

As an operator’s available IPv4 resources are likely to be limited, it may be desirable to use a common range of IPv4 addresses across all of the active Softwire mechanisms. However, this is likely to result in difficulties in routing ingress IPv4 traffic to the correct Border Relay (BR)/AFTR instance which is actively serving a given CE. For example, a client which is configured to use MAP-E may send its traffic to the MAP-E BR, but on the return path, the ingress IP traffic gets routed to a MAP-T BR. The resulting translated packet that gets forwarded to the MAP-E client will be dropped.

Therefore, operators are advised to use separate IPv4 pools for each of the different mechanisms to simplify planning and IPv4 routing.

For IPv6 planning there is less of a constraint as the BR/AFTR elements for the different mechanisms can contain configuration for overlapping client’s IPv6 addresses, providing only one mechanism is actively serving a given client at a time. However, the IPv6 address that is used as the tunnel concentrator’s endpoint (BR/AFTR address) needs to be different for each mechanisms to ensure correct operation.

2.2. Backwards Compatability with Existing Softwire Clients

Deployed clients which can support multiple softwire mechanisms, but do not implement the prioritization mechanism described here may require additional planning. In this scenario, the CPE would request configuration for all of the supported softwire mechanisms in its DHCPv6 Option Request Option (ORO), but would not request OPTION_S46_PRIORITY. By default, the DHCPv6 server will respond with configuration for all of the requested mechanisms which could result in unpredictable and unwanted client configuration.

In this scenario, it may be necessary for the operator to implement logic within the DHCPv6 server to identify such clients and only provision them with configuration for a single softwire mechanism. It should be noted that this can lead to complexity and reduced scalability in the DHCPv6 server implementation due to the addition DHCPv6 message processing overhead.

3. Security Considerations

Security considerations discussed in [RFC6334] and [RFC7598] apply for this document.

Misbehaving intermediate nodes may alter the content of the S46 Priority Option. This may lead to setting a different IPv4 service continuity mechanism than the one initially preferred by the network side. Also, a misbehaving node may alter the content of the S46 Priority Option and other DHCPv6 options (e.g., DHCPv6 Option #64 or #90) so that the traffic is intercepted by an illegitimate node. Those attacks are not unique to the S46 Priority Option but are applicable to any DHCPv6 option that can be altered by a misbehaving intermediate node.

4. IANA Considerations

IANA is kindly requested to allocate the following DHCPv6 option code:

    TBD for OPTION_S46_PRIORITY

All values should be added to the DHCPv6 option code space defined in Section 24.3 of [RFC3315].

4.1. S46 Mechanisms and their Identifying Option Codes

This document requests that IANA create a new registry entitled "Option Codes permitted in the S46 Priority Option". This registry will enumerate the set of DHCPv6 Option Codes that can be included in OPTION_S46_PRIORITY option. Options may be added to this list using the IETF Review process described in Section 4.1 of [RFC5226].

The following table shows the option codes which are currently defined and the S46 mechanisms which they represent. The contents of this table shows the format and the initial values for the new registry. Option codes that have not been requested to be added according to the stated procedure should not be mentioned at all in the table, and should not be listed as "reserved" or "unassigned". The valid range of values for the registry is the range of DHCPv6 Option Codes (1-65535).

Option Code	S46 Mechanism	Reference
64	DS-Lite	[RFC6334]
88	DHCPv4 over DHCPv6	[RFC7341]
94	MAP-E	[RFC7598]
95	MAP-T	[RFC7598]
96	Lightweight 4over6	[RFC7598]

Table 1: DHCPv6 Option to S46 Mechanism Mappings

5. Acknowledgements

Many thanks to O. Troan, S. Barth. A. Yourtchenko, B. Volz, T. Mrugalski, J. Scudder, P. Kyzivat, F. Baker, and B. Campbell for their input and suggestions.

6. References

6.1. 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>.

[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>.

[RFC6334] Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite", RFC 6334, DOI 10.17487/RFC6334, August 2011, <http://www.rfc-editor.org/info/rfc6334>.

[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>.

[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>.

6.2. Informative References

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

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

[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>.

[RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July 2015, <http://www.rfc-editor.org/info/rfc7599>.

Authors' Addresses

Mohamed Boucadair
Orange
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: 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

Copyright Notice

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   C. Xie
Internet-Draft   Q. Sun
Intended status: Standards Track   China Telecom
Expires: September 5, 2014   Q. Sun
                                                     Tsinghua University
                                                                 C. Zhou
                                                     Huawei Technologies
                                                                 T. Tsou
                                              Huawei Technologies (USA)
                                                                  Z. Liu
                                                     Tsinghua University
                                                           March 4, 2014

Radius Extension for Lightweight 4over6
draft-sun-softwire-lw4over6-radext-01

Abstract

lightweight 4over6(lw4over6) [I-D.ietf-softwire-lw4over6] is an extension to DS-Lite in which the amount of state maintained in lwAFTR has been reduced to per-subscriber-level. The lwB4 needs to be provisioned with the public IPv4 address and port set it is allowed to use. The DHCPv4 over DHCPv6 Transport [I.D-ietf-dhc-dhcpv4-over-dhcpv6] and Dynamic Host Configuration Protocol (DHCP) Option for Port Set [I.D-sun-dhc-port-set-option] can be used for lwB4 to provison with the public IPv4 address and port set.

However, in many networks, the configuration information may be stored in Authentication Authorization and Accounting (AAA) servers while user configuration is mainly from Broadband Network Gateway (BNG). This document defines a Remote Authentication Dial In User Service (RADIUS) attribute that carries lightweight 4over6 configuration information from AAA server to BNG.

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 5, 2014.

Copyright Notice

Table of Contents

1. Introduction

Lightweight 4over6 (lw4over6) [I-D.ietf-softwire-lw4over6] defines a model for providing IPv4 access over an IPv6 network in which the Network Address Translation (NAT) function is performed by the Customer-Premises Equipment (CPE) instead of being centralized on a Carrier-Grade NAT (CGN). Lightweight 4over6 features keeping per-subscriber binding state in the service provider's network. This per-subscriber binding state is assigned by the provisioning system and should be synchronized between lwAFTRs. In lw4over6, there are multiple mechanisms to provision an lwB4 with the binding state, including [I.D-ietf-dhc-dhcpv4-over-dhcpv6], [I-D.ietf-softwire-map-dhcp] , or [I-D.ietf-pcp-port-set], etc.

In many networks, user configuration information may be managed by AAA (Authentication, Authorization, and Accounting) servers. Current AAA servers communicate using the Remote Authentication Dial In User Service (RADIUS) [RFC2865] protocol. In a fixed line broadband network, the Broadband Network Gateways (BNGs) act as the access gateway of users. For lw4over6 case, the BNGs are assumed to embed a DHCPv4-over-DHCPv6 server function which allows them to locally handle any DHCPv4-over-DHCPv6 requests issued by hosts. The operators may per-configure subscriber’s binding state in AAA server which then passes the information to a BNG and in turn populates the mapping of the subscribe.

This document defines a new RADIUS attribute that can be used in lightweight 4over6 to carry subscriber’s binding state.

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].

Terminology defined in [I-D.ietf-softwire-lw4over6] is used extensively in this document.

3. Lightweight 4over6 configuration process with RADIUS

The below Figure 1 illustrates how the RADIUS protocol and DHCPv4 -over-DHCPv6 cooperate to provide lwB4 with the binding state.

```
       lwB4                              BNG        lwAFTR          AAA
         |                                |                        Server
         |--PPP LCP Config-Request---->|             |                |
         |                                |             |                |
         |<--PPP LCP Config-ACK  ------|             |                |
         |--PPP IPv6CP Config-Request->|             |                |
         |<-PPP IPv6CP Config-ACK  ----|             |                |
         |-------DHCPv6 Solicit------->|             |                |
         |<------DHCPv6 Advertisement--|             |                |
         |-------DHCPv6 Request------->|             |                |
         |<------DHCPv6 Reply----------|             |                |
         |                                |             |                |
         |--------DHCPv4-QUERY-------> |             |                |
         |    (OPTION_DHCPV4_MSG)      |-------Access-Request------->|
         |                                |     (lw4o6 attr)            |
         |                                |             |<-Configuration-|
         |                                |             |   (Optional)   |
         |                                |             |-----ACK------->|
         |                                |<------Access-Accept---------|
         |                                |     (lw4o6 attr)            |
         |<-------DHCPv4-RESPONSE----- |             |                |
         |     (OPTION_DHCPV4_MSG)     |             |                |
         |                                |             |                |

             DHCPv4-over-DHCPv6           RADIUS
```

Figure 1: Lightweight 4over6 configuration process with RADIUS case 1

BNGs act as a client of RADIUS and as a Unified server. The lwB4 will firstly get the IPv6 address via DHCPv6 process. It then initiates a DHCPv4-QUERY message with OPTION_DHCPV4_MSG Option. Since the lwB4 has known the address of the Unified server in advance, it is recommanded to send the DHCPv4-QUERY message using unicast address. When receving the DHCPv4-QUERY from lwB4, the BNG SHOULD intercept the subscriber’s IPv6 address and stored locally. Then, the BNG SHOULD initiate a RADIUS Access-Request message, in which the User-Name attribute (1) SHOULD be filled by the lwB4 MAC address, to the RADIUS server,the User-password attribute (2) SHOULD be filled by the shared lw4over6 password that has been preconfigured on the DHCPv6 server to get lw4over6 attribute. The IPv6 address in lw4o6 attribute should be filled by the subscriber’s IPv6 address. The AAA server will then determine the IPv4 address and Port Set for the subscriber.

The subscriber’s binding state should be syncronized between AAA server and lwAFTR. If the bindings are pre-configured statically in both AAA server and lwAFTR, the AAA server does not need to configure lwAFTR anymore. Otherwise, if the bindings are locally creately in

AAA server on-demand, it should inform the lwAFTR with the subscriber’s binding state using [I-D.zhou-dime-4over6-provisioning] or COA requests.

Figure 2 illustrates how the RADIUS protocol and DHCPv6 cooperate to provide lwB4 and lwAFTR with tunnel configuration information.

```
    lwB4                 BNG                 AAA Server            lwAFTR
      |                     |                     |                   |
      | --DHCPv6 Request--> |                     |                   |
      |(OPTION_S46_CONT_LW) |                     |                   |
      |                     | --Access-Request--> |                   |
      |                     |    (lw4o6 attr)     |                   |
      |                     |                     |--configuration--> |
      |                     |                     |    (Optional)     |
      |                     |                     | <------ACK------- |
      |                     | <--Access-Accept--- |                   |
      |                     |    (lw4o6 attr)     |                   |
      | <--DHCPv6 Reply---- |                     |                   |
      |(OPTION_S46_CONT_LW) |                     |                   |
             DHCPv6                 Radius
```

Figure 2: Lightweight 4over6 configuration process with RADIUS case 2

BNGs act as a RADIUS client and as a DHCPv6 server. Before the tunnel establishes, lwB4 MAY initiate a DHCPv6 Solicit message that includes an Option Request option[RFC3315] with OPTION_S46_CONT_LW option defined in [I-D.ietf-softwire-map-dhcp]. When BNG receives the SOLICIT, it SHOULD initiates radius Access-Request message, in which the User-Name attribute (1) SHOULD be filled by the lwB4 MAC address, to the RADIUS server,the User-password attribute (2) SHOULD be filled by the shared lw4over6 password that has been preconfigured on the DHCPv6 server to get lw4over6 attribute.

If the authentication request is approved by the AAA server, AAA server will determine the IPv6 address, IPv4 address and Port Set for the subscriber. The subscriber’s binding state should be syncronized between AAA server and lwAFTR. If the bindings are pre-configured statically in both AAA server and lwAFTR, the AAA server does not need to configure lwAFTR anymore. Otherwise, if the bindings are locally creately in AAA server on-demand, it should inform the lwAFTR as mentioned above.

Similarly, BNGs can act as a RADIUS client and as a PCP server in case an lwB4 runs a PCP client (as depicted in Figure 3).

```
      lwB4                           BNG        lwAFTR        AAA Server
       |                              |            |                |
       |---PCP_PORT_SET Request----->|            |                |
       |                              |-------Access-Request------->|
       |                              |     (lw4o6 attr)            |
       |                              |            |<-Configuration-|
       |                              |            |   (Optional)   |
       |                              |            |-----ACK------->|
       |                              |<------Access-Accept---------|
       |                              |     (lw4o6 attr)            |
       |<---PCP_PORT_SET Request----  |            |                |
       |                              |            |                |
       |                              |            |                |

           PCP Port_set           RADIUS
```

Figure 3: Lightweight 4over6 configuration process with RADIUS case 3

In the above-mentioned scenarios, Message-Authenticator (type 80) [RFC2865] SHOULD be used to protect both Access-Request and Access-Accept messages.

After receiving the lw4over6-binding attribute in the initial Access-Accept, the BNG SHOULD store the received lw4over6 configuration parameters locally. When the lw4over6 CE sends a DHCP or PCP Request message to request an extension of the lifetime for the assigned address, the BNG does not have to initiate a new Access-Request towards the AAA server to request the lw4o6 binding state. The BNG could retrieve the previously stored lw4o6 configuration parameters and use them in its reply. The BNG will then inform the AAA server with updated lifetime.

If the BNG does not receive the lw4over6-binding attribute in the Access-Accept or if the BNG receives an Access-Reject, the tunnel cannot be established.

4. Attributes

This section defines the lw4o6_binding attribute that is used in both above-mentioned scenarios. The attribute design follows [RFC6158] and refers to [RFC6929].

4.1. lw4o6_binding Attribute

The lw4o6_binding RADIUS attribute contains the subscriber's binding information including IPv6 address, IPv4 address and the port-set. The BNG SHALL use the binding entry returned in the RADIUS lw4o6_binding attribute to populate the requests.

If the BNG includes the lw4o6_binding attribute, but the AAA server does not recognize it, this attribute MUST be ignored by the AAA server.

If the BNG does not receive the lw4o6_binding attribute in the Access-Accept message and there is the unified server in BNG is not configured to allocate the port-set by itself, the unified SHOULD not response and the tunnel can not be established.

When the Access-Request message is triggered by a DHCP Rebind message, if the binding attribute received in the Access-Accept message is different from the currently used one for that session, the BNG MUST force the lwB4 to re-establish the tunnel using the new binding information received in the Access-Accept message.

The lw4o6_binding Attribute is structured 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     IPv6 address                              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv4 address                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Port Set Index             |        Port Set Mask          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

    Type

      TBD

    Length

      28
    Port Set Index:
         Port Set Index identifies a set of ports assigned
         to a device.  The first k bits on the left of the 2-octet
         field is the Port Set Index value, with the rest of the
         field right padding zeros.
    Port Set Mask:
         Port Set Mask indicates the position of the bits
         used to build the mask.  The first k bits on the left is
         padding ones while the remained (16-k) bits of the 2-octet
         field on the right is padding zeros.
    IPv4 address
      The translated IPv4 address for a subscriber.
    IPv6 address
      The IPv6 address for a subscriber.
```

Figure 4: Lightweight 4over6 Attribute

5. Table of attributes

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

Request	Accept	Reject	Challenge	Accounting Request	#	Attribute
0-1	0-1	0	0	0-1	TBD1	lw4o6-binding
0-1	0-1	0	0	0-1	1	User-Name
0-1	0	0	0	0	2	User-Password
0-1	0-1	0	0	0-1	6	Service-Type
0-1	0-1	0-1	0-1	0-1	80	Message-Authenticator

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

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

Figure 5: Lightweight 4over6 Attribute Table

6. Security Considerations

TO BE COMPLETED

7. IANA Considerations

This document has no IANA actions.

8. Acknowledgements

The authors would like to thank the following individuals who have participated in the drafting, review, and discussion of this memo: TO BE COMPLETED

9. References

9.1. Normative References

[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-00 (work in progress), March 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.

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

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

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

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

9.2. Informative References

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

Authors’ Addresses

Chongfeng Xie
China Telecom
P.R.China

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

Qiong Sun
China Telecom
P.R.China

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

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

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

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

Email: cathy.zhou@huawei.com

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

Phone: +1 408 330 4424
Email: Tina.Tsou.Zouting@huawei.com

ZiLong Liu
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: liuzilong8266@126.com

Softwire Working Group Q. Sun
Internet-Draft H. Wang
Intended status: Standards Track Y. Cui
Expires: January 9, 2017 Tsinghua University
I. Farrer
S. Zoric
Deutsche Telekom AG
M. Boucadair
Orange
R. Asati
Cisco Systems, Inc.
July 8, 2016

A YANG Data Model for IPv4-in-IPv6 Softwires
draft-sun-softwire-yang-05

Abstract

This document defines a YANG data model for the configuration and operations (state, notification, RPC etc.) of IPv4-in-IPv6 Softwire Border Routers and Customer Premises Equipment. The model covers the Lightweight 4over6, MAP-E and MAP-T Softwire mechanisms.

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

The IETF Softwire Working Group has developed several IPv4-in-IPv6 Softwire mechanisms to address various deployment contexts and constraints. As a companion to the architectural specification documents, this document focuses on the provisioning of A+P softwire functional elements: Border Routers (BRs) and Customer Premises Equipment (CEs).

This document defines a YANG data model [RFC6020] that can be used to configure and manage A+P Softwire elements using the NETCONF protocol [RFC6241]. DS-Lite YANG data model is defined in [I-D.boucadair-softwire-dslite-yang].

The Softwire YANG model is structured into two sub-models:

o Lightweight 4over6 [RFC7596]

o MAP-E [RFC7597] and MAP-T [RFC7599] (combined due to their common configuration parameters).

Two root containers are defined:

1. Container "softwire-config" holds the collection of YANG definitions common to all Softwire element configuration.

2. Container "softwire-state" holds YANG definitions for the operational state of the Softwire elements.

A NETCONF notify module is also included.

This approach has been taken so that the model can be easily extended to support additional Softwire mechanisms, if required.

1.1. Terminology

The reader should be familiar with the concepts and terms defined in [RFC7596], [RFC7597], [RFC7599], and the YANG data modelling language [RFC6020].

1.2. Tree Diagrams

The meaning of the symbols in these diagrams are as follows:

o Brackets "[" and "]" enclose list keys.

o Braces "{" and "}" enclose feature content.

o Parentheses "(" and ")" enclose choice and case nodes, and case nodes are also marked with a colon (":").

o Symbols after data node names: "?" means an optional node, and "*" denotes a list and leaf-list.

o Abbreviations before data node names: "rw" means configuration data (read-write), and "ro" means state data (read-only).

1.3. YANG Modelling of NAT44 Functionality

The model does not include CPE NAT-specific provisioning parameters that may be used for IPv4 address sharing other than the external IP address and port set which a softwire client may use for NAT44. NAT-specific considerations are out of scope of this document. A YANG model for the configuration and management of NAT gateways is described in [I-D.sivakumar-yang-nat].

2. Common

The following sections of the document are structured with the root of the Softwire YANG model (common to all mechanisms) described first. Subsequent sections describe the models relevant to the different softwire mechanisms. All functions are listed, but the YANG models use the "feature" statement to distinguish among the different softwire mechanisms. This document defines a new module named "ietf-softwire" for Softwire data models such that this module aments "ietf-ipv6-unicast-routing" module that is defined in [I-D.ietf-netmod-routing-cfg].

3. Lightweight 4over6

Lightweight 4over6 (binding) includes two elements: lwAFTR (BR) and lwB4 (CE). The lwAFTR holds configuration for IPv4-IPv6 address bindings which are used for the forwarding of traffic originating from lwB4s.

The lwB4 is configured with the relevant parameters for establishing the IPv4-in-IPv6 tunnel including an IPv6 address for the lwAFTR and the IPv4 configuration for NAT44.

4. MAP-E and MAP-T

MAP-E and MAP-T elements are provisioned with the MAP rules necessary for defining MAP domains and forwarding rules. For MAP-T CEs, an additional "ipv6-prefix" parameter is also included. Note that when referring to MAP-E/T (algorithm), the CE and BR shares the same model for configuration and management.

5. Softwire YANG Tree Diagrams

5.1. Common Tree Diagrams

Figure 1 describes the high level softwire YANG data model and the way tree is organized is common to all of the different softwire mechanisms listed in Section 1:

```
      +--rw softwire-config
      |  +--rw description?              string
      |  +--rw binding {binding}?
      |  |  +--rw br {br}?
      |  |  +--rw cr {cr}?
      |  +--rw algorithm {algorithm}?
      |
      +--ro softwire-state
         +--ro description?              string
         +--ro binding {binding}?
         |  +--ro br {br}?
         |  +--ro ce {ce}?
         +--ro algorithm {algorithm}?
```

Figure 1: High Level Softwire YANG Tree Organization

5.2. Lightweight 4over6 Tree Diagrams

Figure 2 defines the softwire data model for lw4o6 (softwire binding mode) which includes lwAFTR (BR) and lwB4 (CE):

```
module: ietf-softwire
   +--rw softwire-config
   |  +--...
   |  +--rw binding {binding}?
   |     +--rw br {br}?
   |     |  +--rw enable?                       boolean
   |     |  +--rw br-instances
   |     |     +--rw br-instance* [id]
   |     |        +--rw binding-table-versioning
   |     |        |  +--rw binding-table-version?  uint64
   |     |        |  +--rw binding-table-date?     yang:date-and-time
   |     |        +--rw id                         uint32
   |     |        +--rw name?                      string
   |     |        +--rw softwire-num-threshold     uint32
   |     |        +--rw tunnel-payload-mtu         uint16
   |     |        +--rw tunnel-path-mru            uint16
   |     |        +--rw binding-table
   |     |           +--rw binding-entry* [binding-ipv6info]
   |     |              +--rw binding-ipv6info     union
   |     |              +--rw binding-ipv4-addr    inet:ipv4-address
   |     |              +--rw port-set
   |     |              |  +--rw psid-offset       uint8
   |     |              |  +--rw psid-len          uint8
   |     |              |  +--rw psid              uint16
   |     |              +--rw br-ipv6-addr         inet:ipv6-address
   |     |              +--rw lifetime?            uint32
   |     +--rw ce {ce}?
```

```
   |        +--rw enable?                         boolean
   |        +--rw ce-instances
   |           +--rw ce-instance* [binding-ipv6info]
   |              +--rw name?                     string
   |              +--rw tunnel-payload-mtu        uint16
   |              +--rw tunnel-path-mru           uint16
   |              +--rw b4-ipv6-addr-format       boolean
   |              +--rw binding-ipv6info          union
   |              +--rw binding-ipv4-addr         inet:ipv4-address
   |              +--rw port-set
   |              |  +--rw psid-offset            uint8
   |              |  +--rw psid-len               uint8
   |              |  +--rw psid                   uint16
   |              +--rw br-ipv6-addr              inet:ipv6-address
   |              +--rw lifetime?                 uint32
   +--ro softwire-state
      +--...
      +--ro binding {binding}?
         +--ro br {br}?
         |  +--ro br-instances
         |     +--ro br-instance* [id]
         |        +--ro id                        uint32
         |        +--ro name?                     string
         |        +--ro sentPacket?               yang:zero-based-counter64
         |        +--ro sentByte?                 yang:zero-based-counter64
         |        +--ro rcvdPacket?               yang:zero-based-counter64
         |        +--ro rcvdByte?                 yang:zero-based-counter64
         |        +--ro droppedPacket?            yang:zero-based-counter64
         |        +--ro droppedByte?              yang:zero-based-counter64
         |        +--ro active-softwire-num?      uint32
         |        +--ro binding-table
         |           +--ro binding-entry* [binding-ipv6info]
         |              +--ro binding-ipv6info    union
         |              +--ro active?             boolean
         +--ro ce {ce}?
            +--ro ce-instances
               +--ro ce-instance* [binding-ipv6info]
                  +--ro name?                     string
                  +--ro binding-ipv6info          union
                  +--ro sentPacket?               yang:zero-based-counter64
                  +--ro sentByte?                 yang:zero-based-counter64
                  +--ro rcvdPacket?               yang:zero-based-counter64
                  +--ro rcvdByte?                 yang:zero-based-counter64
                  +--ro droppedPacket?            yang:zero-based-counter64
                  +--ro droppedByte?              yang:zero-based-counter64
```

Figure 2: Softwire Lightweight 4over6 Data Model Tree Structure

The data model assumes that each CE/BR instance can: be enable/disabled, be provisioned with a dedicated configuration data, and maintain its own binding table.

Additional information on some of the important lwAFTR nodes is provided below:

- binding-table-versioning: optionally used to add a incremental version number and/or timestamp to the binding table. This can be used for logging/data retention purposes. The version number is incremented and a new timestamp value written whenever a change is made to the contents of the binding table or a new binding table list is created.

- binding-entry: used to define the binding relationship between 3-tuples, which contains the lwB4's IPv6 address/prefix, the allocated IPv4 address and restricted port-set. For detail information, please refer to [RFC7596].

- tunnel-payload-mtu: used to set the IPv4 MTU for the lw4o6 tunnel.

- tunnel-path-mru: used to set the maximum lw4o6 IPv6 encapsulating packet size that can be received.

- psid-offset: used to set the number of offset bits.

- psid-len: defines the number of ports that will be allocated for the softwire.

- psid: used to identify the set of ports allocated for a specific softwire.

- tunnel-num-threshold: used to set the maximum number of tunnels that can be created on the lw4o6 device simultaneously.

- active-tunnel-num (ro): used to present the number of tunnels currently provisioned on the device.

- active (ro): used to show the status of particular binding-entry.

Additional information on some of the important lwB4 nodes is provided below:

- b4-ipv6-addr-format: indicates the format of lwB4 IPv6 address. If set to true, it indicates that the IPv6 source address of the lwB4 is constructed according to the description in Section 6 of [RFC7597]; if set to false, the lwB4 can use any /128 address from the assigned IPv6 prefix.

o binding-ipv6info: used to set the IPv6 address type which is combined in a binding entry, for a complete address or a prefix.

5.3. MAP-E and MAP-T Tree Diagrams

Figure 3 defines the softwire data model for MAP-E and MAP-T:

```
module: ietf-softwire
   +--rw softwire-config
   |  +--...
   |  +--rw algorithm {algorithm}?
   |     +--rw enable?                          boolean
   |     +--rw algorithm
   |        +--rw algo-instance* [id]
   |           +--rw algo-versioning
   |           |  +--rw algo-version?           uint64
   |           |  +--rw algo-date?              yang:date-and-time
   |           +--rw id                         uint32
   |           +--rw name?                      string
   |           +--rw data-plane                 enumeration
   |           +--rw ea-len                     uint8
   |           +--rw rule-ipv6-prefix           inet:ipv6-prefix
   |           +--rw rule-ipv4-prefix           inet:ipv4-prefix
   |           +--rw forwarding                 boolean
   |           +--rw psid-offset                uint8
   |           +--rw psid-len                   uint8
   |           +--rw tunnel-payload-mtu         uint16
   |           +--rw tunnel-path-mru            uint16
   |           +--rw br-ipv6-addr               inet:ipv6-address
   |           +--rw dmr-ipv6-addr              inet:ipv6-prefix
   +--ro softwire-state
      +--...
      +--ro algorithm {algorithm}?
         +--ro algo-instances
            +--ro algo-instance* [id]
               +--ro id                         int32
               +--ro name?                      string
               +--ro sentPacket?                yang:zero-based-counter64
               +--ro sentByte?                  yang:zero-based-counter64
               +--ro rcvdPacket?                yang:zero-based-counter64
               +--ro rcvdByte?                  yang:zero-based-counter64
               +--ro droppedPacket?             yang:zero-based-counter64
               +--ro droppedByte?               yang:zero-based-counter64
```

Figure 3: Softwire MAP-E and MAP-T Data Model Structure

Additional information on some of the important MAP-E and MAP-T nodes is provided below:

o algo-versioning: optionally used to add a incremental version number and/or timestamp to the algorithm. This can be used for logging/data retention purposes. The version number is incremented and a new timestamp value written whenever a change is made to the algorithm or a new instance is created.

o forwarding: specifies whether the rule can be used as a Forward Mapping Rule (FMR). If not set, this rule is a Basic Mapping Rule (BMR) only and must not be used for forwarding. See Section 4.1 of [RFC7598].

o ea-len: used to set the length of the Embedded-Address (EA), which defined in the mapping rule for a MAP domain.

o dmr-ipv6-prefix: defines the Default Mapping Rule (DMR) for MAP-T. This parameter is optional when configuring a MAP-T BR.

o stat-count (ro): use to show the numbers of packets and bytes information of specific device respectively.

5.4. Notifications for Softwire YANG

This section describes the tree structure for notifications. These notifications pertain to the configuration and monitoring portions of the specific Softwire mechanisms. The logic is that the softwire instance notifies the NETCONF client with the index for a mapping entry and the NETCONF client retrieves the related information from the operational datastore of that instance.

```
module: ietf-softwire
notifications:
   +---n softwire-binding-br-event {binding,br}?
   |  +--ro br-id?            -> /softwire-state/binding/br/.../id
   |  +--ro invalid-entry*    -> /softwire-config/binding/br/.../binding-table/b
inding-entry/binding-ipv6info
   |  +--ro added-entry*      inet:ipv6-address
   |  +--ro modified-entry*   -> /softwire-config/binding/br/.../binding-table/b
inding-entry/binding-ipv6info
   +---n softwire-binding-ce-event {binding,ce}?
   |  +--ro ce-binding-ipv6-addr-change    inet:ipv6-address
   +---n softwire-algorithm-instance-event {algorithm}?
      +--ro algo-id           -> /softwire-config/algorithm/.../id
      +--ro invalid-entry*    -> /softwire-config/algorithm/.../id
      +--ro added-entry*      -> /softwire-config/algorithm/.../id
      +--ro modified-entry*   -> /softwire-config/algorithm/.../id
```

Figure 4: Softwire Notifications Data Model Structure

Additional information on some of the important notification nodes is listed below:

o invalid-entry, added-entry, modified-entry: used to notify the client that a specific binding entry or MAP rule is expired or invalidated, added, or modified.

o ce-binding-ipv6-addr-change: used to notify that the lwB4's binding-ipv6-address has been changed or the value of the 'b4-ipv6-addr-format' is "False".

6. Softwire YANG Model

This module imports typedefs from [RFC6991].

```
<CODE BEGINS> file "ietf-softwire@2016-06-04.yang"

module ietf-softwire {
  namespace "urn:ietf:params:xml:ns:yang:ietf-softwire";
  prefix "softwire";

  import ietf-inet-types {prefix inet; }
  import ietf-yang-types {prefix yang; }

  organization "Softwire Working Group";

  contact
    "
    Qi Sun <sunqi.ietf@gmail.com>
    Hao Wang <wangh13@mails.tsinghua.edu.cn>
    Yong Cui <yong@csnet1.cs.tsinghua.edu.cn>
    Ian <Farrer ian.farrer@telekom.de>
    Sladjana Zoric <sladjana.zoric@telekom.de>
    Mohamed Boucadair <mohamed.boucadair@orange.com>
    Rajiv <Asati rajiva@cisco.com>
    ";

  description
    "This document defines a YANG data model for the configuration and
    management of A+P Softwire Border Routers (BRs) and Customer
    Premises Equipment (CEs). It covers Lightweight 4over6,
    MAP-E and MAP-T mechanisms.

    Copyright (c) 2016 IETF Trust and the persons identified
    as authors of the code. All rights reserved.
    This version of this YANG module is part of RFC XXX; see the RFC
    itself for full legal notices.";

  revision 2016-06-04 {
    description
      "Version-05: Combined MAP-E/MAP-T into a single tree. Added binding
```

```
          table/alogorthm versioning";
       reference "-05";
  }

  revision 2015-09-30 {
    description
      "Version-04: Fix YANG syntax; Add flags to map-rule; Remove
      the map-rule-type element. ";
       reference "-04";
  }

  revision 2015-04-07 {
    description
      "Version-03: Integrate lw4over6; Updata state nodes; Correct
      grammar errors; Reuse groupings; Update descriptions.
      Simplify the model.";
       reference "-03";
  }

  revision 2015-02-10 {
    description
      "Version-02: Add notifications.";
       reference "-02";
  }


  revision 2015-02-06 {
    description
      "Version-01: Correct grammar errors; Reuse groupings; Update
      descriptions.";
       reference "-01";
  }

  revision 2015-02-02 {
    description
      "Initial revision.";
       reference "-00";
  }

/*
 * Features
 */

  feature binding {
    description
    "Lightweight 4over6 (binding) is an IPv4-over-IPv6 tunnelling
     transition mechanism. Lightweight 4over6 is a solution designed
     specifically for complete independence between IPv6 subnet
```

```
     prefix (and /128 IPv6 address) and IPv4 address with or
     without IPv4 address sharing.

     This is accomplished by maintaining state for
     each softwire (per-subscriber state) in the central lwAFTR and
     a hub-and-spoke forwarding architecture. In order to delegate
     the NAPT function and achieve IPv4 address sharing,
     port-restricted IPv4 addresses needs to be allocated to CEs.

     Besides lw4o6, this feature also covers MAP in 1:1 mode
     (offset=0, PSID explicit)";

   reference
      "RFC7596";
  }

  feature br {
    if-feature binding;
    description
     "The AFTR for Lightweight 4over6, so-called lwAFTR (BR). This
      feature indicates that a instance functions as a lwAFTR (BR).
      A lwAFTR (BR) is an IPv4-in-IPv6 tunnel concentrator that
      maintains per-subscriber IPv4-IPv6 address binding.";
  }

  feature ce {
    if-feature binding;
    description
      "The B4 for Lightweight 4over6, so-called lwB4 (CE). This
      feature indicates that a instance functions as a lwB4 (CE). A
      lwB4 (ce) is an IPv4-in-IPv6 tunnel initiator. It is
      dual-stack capable node, either a directly connected end-host
      or a CE. It sources IPv4 connections using the configured
      port-set and the public IPv4 address.";
  }

  feature algorithm {
    description
      "MAP-E is an IPv6 transition mechanism for transporting IPv4
      packets across an IPv6 network using IP encapsulation. MAP-E
      allows for a reduction of the amount of centralized state using
      rules to express IPv4/IPv6 address mappings. This introduces an
      algorithmic relationship between the IPv6 subnet
      and IPv4 address.
      The Mapping of Address and Port - Translation (MAP-T)
      architecture is a double stateless NAT64 based solution. It uses
      the stateless algorithmic address & transport layer port mapping
      scheme defined in MAP-E. The MAP-T solution differs from MAP-E in
```

```
      the use of IPv4-IPv6 translation, rather than encapsulation, as
      the form of IPv6 domain transport.
      This feature indicates the instance functions as a MAP-E or
      MAP-T instance.";
    reference
      "RFC7597 & RFC7599";
  }

/*
 * Grouping
 */

  grouping port-set {
    description
      "Use the PSID algorithm to represent a range of transport layer
      ports.";
    leaf psid-offset {
      type uint8 {
        range 0..16;
      }
      mandatory true;
      description
        "The number of offset bits. In Lightweight 4over6, the default
        value is 0 for assigning one contiguous port range. In MAP-E/T,
        the default value is 6, which excludes system ports by default
        and assigns distributed port ranges. If the this parameter is
        larger than 0, the value of offset MUST be greater than 0.";
    }
    leaf psid-len {
      type uint8 {
        range 0..15;
      }
      mandatory true;
      description
        "The length of PSID, representing the sharing ratio for an
        IPv4 address.";
    }
    leaf psid {
      type uint16;
      mandatory true;
      description
        "Port Set Identifier (PSID) value, which identifies a set
        of ports algorithmically.";
    }
  }

  grouping binding-entry {
    description
```

```
        "The lwAFTR maintains an address binding table that contains
        the binding between the lwB4's IPv6 address, the allocated IPv4
        address and restricted port-set.";
      leaf binding-ipv6info {
        type union {
          type inet:ipv6-address;
          type inet:ipv6-prefix;
        }
        mandatory true;
        description
          "The IPv6 information for a binding entry.
           If it's an IPv6 prefix, it indicates that
           the IPv6 source address of the lwB4 is constructed
           according to the description in RFC7596;
           if it's an IPv6 address, it means the lwB4 uses
           any /128 address from the assigned IPv6 prefix.
           ";
      }
      leaf binding-ipv4-addr {
        type inet:ipv4-address;
        mandatory true;
        description
          "The IPv4 address assigned to the lwB4, which is
           used as the IPv4 external address
           for lwB4 local NAPT44.";
      }
      container port-set {
        description
          "For Lightweight 4over6, the default value
          of offset should be 0, to configure one contiguous
          port range.";
        uses port-set {
          refine psid-offset {
            default "0";
          }
        }
      }
      leaf br-ipv6-addr {
        type inet:ipv6-address;
        mandatory true;
        description
          "The IPv6 address for lwaftr.";
      }
      leaf lifetime {
        type uint32;
        units seconds;
        description "The lifetime for the binding entry";
      }
```

```
  }

/*
  grouping nat-table {

    description
      "Grouping 'nat-table' is not extended. The current mechanism
      is focusing on the provisioning of external IP address and
      port set; other NAT-specific considerations are out of scope.";
  }
*/

  grouping traffic-stat {
    description "Traffic statistics";
    leaf sentPacket {
      type yang:zero-based-counter64;
      description "Number of packets sent.";
    }
    leaf sentByte {
      type yang:zero-based-counter64;
      description "Traffic sent, in bytes";
    }
    leaf rcvdPacket {
      type yang:zero-based-counter64;
      description "Number of packets received.";
    }
    leaf rcvdByte {
      type yang:zero-based-counter64;
      description "Traffic received, in bytes";
    }
    leaf droppedPacket {
      type yang:zero-based-counter64;
      description "Number of packets dropped.";
    }
    leaf droppedByte {
      type yang:zero-based-counter64;
      description "Traffic dropped, in bytes";
    }
  }


/*
 * Configuration Data Nodes
 */


  container softwire-config {
    description
```

```
      "The configuration data for Softwire instances. And the shared
      data describes the softwire data model which is common to all of
      the different softwire mechanisms, such as description.";
    leaf description {
      type string;
      description
        "A textual description of Softwire.";
    }
    container binding {
      if-feature binding;
      description
        "lw4over6 (binding) configuration.";
      container br {
        if-feature br;
        description
          "Indicate this instance supports the lwAFTR (BR) function.
          The instances advertise the BR feature through the
          capability exchange mechanism when a NETCONF session is
          established.";
        leaf enable {
          type boolean;
          description
            "Enable/disable the lwAFTR (BR) function.";
        }
        container br-instances {
          description
            "A set of BRs to be configured.";
          list br-instance {
            key "id";
            description
            "A set of lwAFTRs to be configured.";
            container binding-table-version {
              description "binding table's version";
              leaf binding-table-version{
                type uint64;
                description "Incremental version number
                to the binding table";
              }
              leaf binding-table-date {
                type yang:date-and-time;
                description "Timestamp to the binding
                table";
              }
            }
            leaf id {
              type uint32;
              mandatory true;
              description "An instance identifier.";
```

```
            }
            leaf name {
              type string;
              description "The name for the lwaftr.";
            }
            leaf softwire-num-threshold {
              type uint32;
              mandatory true;
              description
                "The maximum number of tunnels that can be created on
                the lwAFTR.";
            }
            leaf tunnel-payload-mtu {
              type uint16;
              mandatory true;
              description
                "The payload MTU for Lightweight 4over6 tunnel.";
            }
            leaf tunnel-path-mru {
              type uint16;
              mandatory true;
              description
                "The path MRU for Lightweight 4over6 tunnel.";
            }
            container binding-table {
              description "binding table";
              list binding-entry {
                key "binding-ipv6info";
                description "binding entry";
                uses binding-entry;
              }
            }
          }
        }
      }

      container ce {
        if-feature ce;
        description
          "Indicate this instance supports the lwB4 (CE) function.
          The instances advertise the CE feature through the
          capability exchange mechanism when a NETCONF session is
          established.";
        leaf enable {
          type boolean;
          description
            "Enable/disable the lwB4 (CE) function.";
        }
```

```
        container ce-instances {
          description
            "A set of CEs to be configured.";
          list ce-instance {
            key "binding-ipv6info";
            description "instances for CE";
            leaf name {
              type string;
              description "The CE’s name.";
            }
            leaf tunnel-payload-mtu {
              type uint16;
              mandatory true;
              description
                "The payload MTU for Lightweight 4over6 tunnel.";
            }
            leaf tunnel-path-mru {
              type uint16;
              mandatory true;
              description
                "The path MRU for Lightweight 4over6 tunnel.";
            }
            leaf b4-ipv6-addr-format {
              type boolean;
              mandatory true;
              description
               "The format of lwB4 (CE) IPv6 address. If set to true,
               it indicates that the IPv6 source address of the lwB4
               is constructed according to the description in
               [RFC7596]; if set to false, the lwB4 (CE)
               can use any /128 address from the assigned IPv6
               prefix.";
            }
            uses binding-entry;
          }
        }
      }
    }

    container algorithm {
      if-feature algorithm;
      description
        "Indicate the instances support the MAP-E and MAP-T function.
        The instances advertise the map-e feature through the
        capability exchange mechanism when a NETCONF session is
        established.";
      leaf enable {
        type boolean;
```

```
        description
          "Enable/disable the MAP-E or MAP-T function.";
      }
      container algo-instances {
        description
          "A set of MAP-E or MAP-T instances to be configured,
           applying to BRs and CEs. A MAP-E/T instance defines a MAP
           domain comprising one or more MAP-CE and MAP-BR";
        list algo-instance {
          key "id";
          description "instance for MAP-E/MAP-T";
          container algo-versioning {
            description "algorithm's version";
            leaf algo-version {
              type uint64;
              description "Incremental version number to
              the algorithm";
            }
            leaf algo-date {
              type yang:date-and-time;
              description "Timestamp to the algorithm";
            }
          }
          leaf id {
            type uint32;
            mandatory true;
            description "Algorithm Instance ID";
          }
          leaf name {
            type string;
            description "The name for the instance.";
          }
          leaf data-plane {
            type enumeration {
              enum "encapsulation" {
                description "encapsulation for MAP-E";
              }
              enum "translation" {
                description "translation for MAP-T";
              }
            }
            description
              "Encapsulation is for MAP-E while translation is
              for MAP-T";
          }
          leaf ea-len {
            type uint8;
            mandatory true;
```

```
          description
            "Embedded Address (EA) bits are the IPv4 EA-bits
            in the IPv6 address identify an IPv4
            prefix/address (or part thereof) or
            a shared IPv4 address (or part thereof)
            and a port-set identifier.
            The length of the EA-bits is defined as
            part of a MAP rule for a MAP domain.";
        }
        leaf rule-ipv6-prefix {
          type inet:ipv6-prefix;
          mandatory true;
          description
            "The Rule IPv6 prefix defined in the mapping rule.";
        }
        leaf rule-ipv4-prefix {
          type inet:ipv4-prefix;
          mandatory true;
          description
            "The Rule IPv4 prefix defined in the mapping rule.";
        }
        leaf forwarding {
          type boolean;
          mandatory true;
          description
            "This parameter specifies whether the rule may be used for
            forwarding (FMR). If set, this rule is used as an FMR;
            if not set, this rule is a BMR only and must not be used
            for forwarding.";
        }
        leaf psid-offset {
          type uint8 {
            range 0..16;
          }
          mandatory true;
          description
            "The number of offset bits. In Lightweight 4over6, the default
            value is 0 for assigning one contiguous port range. In MAP-E/T,
            the default value is 6, which excludes system ports by default
            and assigns distributed port ranges. If the this parameter is
            larger than 0, the value of offset MUST be greater than 0.";
        }
        leaf psid-len {
          type uint8 {
            range 0..15;
          }
          mandatory true;
          description
```

```
              "The length of PSID, representing the sharing ratio for an
              IPv4 address.";
          }
          leaf tunnel-payload-mtu {
            type uint16;
            description
              "The payload MTU for MAP-E tunnel.";
          }
          leaf tunnel-path-mru {
            type uint16;
            description
              "The path MRU for MAP-E tunnel.";
          }
          leaf br-ipv6-addr {
            type inet:ipv6-address;
            mandatory true;
            description
              "The IPv6 address of the MAP-E BR.";
          }
          leaf dmr-ipv6-prefix {
            type inet:ipv6-prefix;
            description
              "The IPv6 prefix of the MAP-T BR. ";
          }
        }
      }
    }
  }

/*
 * Operational state Data Nodes
 */

  container softwire-state {
    config false;
    description
      "The operational state data for Softwire instances. ";
    leaf description {
      type string;
      description
        "A textual description of the softwire instances.";
    }
    container binding {
      if-feature binding;
      description
        "lw4over6 (binding) state.";
      container br {
        if-feature br;
```

```
        config false;
        description
          "Indicate this instance supports the lwAFTR (BR) function.
          The instances advertise the lwaftr (BR) feature through the
          capability exchange mechanism when a NETCONF session is
          established.";
        container br-instances {
          description
            "A set of BRs.";
          list br-instance {
            key "id";
            description "instances for BR";
            leaf id {
              type uint32;
              mandatory true;
              description "id";
            }
            leaf name {
              type string;
              description "The name for this lwaftr.";
            }
            uses traffic-stat;
            leaf active-softwire-num {
              type uint32;
              description
                "The number of currently active tunnels on the
                lw4over6 (binding) instance.";
            }
            container binding-table {
              description "id";
              list binding-entry {
                key "binding-ipv6info";
                description "An identifier of the binding entry.";
                leaf binding-ipv6info {
                  type union {
                    type inet:ipv6-address;
                    type inet:ipv6-prefix;
                  }
                  mandatory true;
                  description
                    "The IPv6 information used to identify
                      a binding entry. ";
                }
                leaf active {
                  type boolean;
                  description
                    "Status of a specific tunnel.";
                }
```

```
              }
            }
          }
        }
      }


      container ce {
        if-feature ce;
        config false;
        description
          "Indicate this instance supports the lwB4 (CE) function.
          The instances advertise the lwb4 (CE) feature through the
          capability exchange mechanism when a NETCONF session is
          established.";
        container ce-instances {
          description
            "Status of the configured CEs.";
          list ce-instance {
            key "binding-ipv6info";
            description "a lwB4 (CE) instance.";
            leaf name {
              type string;
              description "The CE’s name.";
            }
            leaf binding-ipv6info {
              type union {
                type inet:ipv6-address;
                type inet:ipv6-prefix;
              }
              mandatory true;
              description
                "The IPv6 information used to identify
                 a binding entry. ";
            }
            uses traffic-stat;
          }
        }
      }
    }

    container algorithm {
      if-feature algorithm;
      config false;
      description
        "Indicate the instances support the MAP-E and MAP-T function.
        The instances advertise the map-e/map-t feature through the
        capability exchange mechanism when a NETCONF session is
```

```
        established.";
      container algo-instances {
        description
          "Status of MAP-E instance(s).";
        list algo-instance {
          key "id";
          description "Instances for algorithm";
          leaf id {
            type uint32;
            mandatory true;
            description "id";
          }
          leaf name {
            type string;
            description "The map-e instance name.";

          }
          uses traffic-stat;
        }
      }
    }
  }

/*
 * Notifications
 */
  notification softwire-br-event {
    if-feature binding;
    if-feature br;
    description "Notification for BR.";

    leaf br-id {
      type leafref {
        path
          "/softwire-state/binding/br/br-instances/"
          + "br-instance/id";
      }
          description "...";
    }
    leaf-list invalid-entry {
      type leafref {
        path
          "/softwire-config/binding/br/br-instances/"
          + "br-instance[id=current()/../br-id]/"
          + "binding-table/binding-entry/binding-ipv6info";
      }
      description
        "Notify the client that a specific binding entry has been
```

```
        expired/invalid. The binding-ipv6info identifies an entry.";
    }
    leaf-list added-entry {
        type inet:ipv6-address;
        description
          "Notify the client that a binding entry has been added.
          The ipv6 address of that entry is the index. The client
          get other information from the lwaftr about the entry
          indexed by that ipv6 address.
          ";
    }
    leaf-list modified-entry {
        type leafref {
          path
            "/softwire-config/binding/br/br-instances/"
            + "br-instance[id=current()/../br-id]/"
            + "binding-table/binding-entry/binding-ipv6info";
        }
          description "...";
    }
  }

  notification softwire-ce-event {
    if-feature binding;
    if-feature ce;
    description "CE notification";
    leaf ce-binding-ipv6-addr-change {
      type inet:ipv6-address;
      mandatory true;
      description
        "The source tunnel IPv6 address of the lwB4.
         If 'b4-ipv6-addr-format' is false, or the lwb4's
         binding-ipv6-address changes for any reason,
         it SHOULD notify the NETCONF client.";
    }
  }

  notification softwire-algorithm-instance-event {
    if-feature algorithm;
    description "Notifications for MAP-E or MAP-T.";
    leaf algo-id {
      type leafref {
        path
          "/softwire-config/algorithm/algo-instances/algo-instance/id";
      }
      mandatory true;
      description "MAP-E or MAP-T event.";
    }
```

```
    leaf-list invalid-entry-id {
      type leafref {
        path
          "/softwire-config/algorithm/algo-instances/algo-instance/id";
      }
      description "Invalid entry event.";
    }
    leaf-list added-entry {
      type leafref {
        path
          "/softwire-config/algorithm/algo-instances/algo-instance/id";
      }
      description "Added entry.";
    }
    leaf-list modified-entry {
      type leafref {
        path
          "/softwire-config/algorithm/algo-instances/algo-instance/id";
      }
      description "Modified entry.";
    }
  }
}
<CODE ENDS>
```

7. Example of Configure lw4o6 Binding-Table

The lwAFTR maintains an address binding table which contains the following 3-tuples:

- o IPv6 Address for a single lwB4
- o Public IPv4 Address
- o Restricted port-set

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

Let’s consider an example to add an entry that maintains the relationship between 3-tuples of lwB4 (2001:db8::1), ’192.0.2.1’ and ’1234’ in the binding table of the lwAFTR (2001:db8::2). Here is the example binding-table configuration xml:

```
<rpc message-id="101"
  xmlns:nc="urn:params:xml:ns:yang:ietf-softwire:1.0">
<!-- replace with IANA namespace when assigned. -->
  <edit-config>
    <target>
      <running/>
    </target>
  <softwire-config>
    <lw4o6-aftr>
      <lw4o6-aftr-instances>
        <lw4o6-aftr-instance>
          <aftr-ipv6-addr>2001:db8::2</aftr-ipv6-addr>
          <binding-table>
            <binding-entry>
              <binding-ipv4-addr>192.0.2.1</binding-ipv4-addr>
              <port-set>
                <psid>1234</psid>
              </port-set>
              <binding-ipv6-addr>2001:db8::1</binding-ipv6-addr>
              <active>1</active>
            </binding-entry>
          </binding-table>
        </lw4o6-aftr-instance>
      </lw4o6-aftr-instances>
    </lw4o6-aftr>
  </softwire-config>
```

Figure 5: lw4o6 Binding-Table Configuration XML

8. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory to implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

All data nodes defined in the YANG module which can be created, modified and deleted (i.e., config true, which is the default). These data nodes are considered sensitive. Write operations (e.g., edit-config) applied to these data nodes without proper protection can negatively affect network operations.

9. IANA Considerations

This document requests IANA to register the following URI in the "IETF XML Registry" [RFC3688].

URI: urn:ietf:params:xml:ns:yang:softwire
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

This document requests IANA to register the following YANG module in the "YANG Module Names" registry [RFC6020].

name: ietf-dslite-aftr
namespace: urn:ietf:params:xml:ns:yang:softwire
prefix: softwire
reference: RFC XXXX

10. Acknowledgements

The authors would like to thank Lishan Li, Bert Wijnen, Giles Heron, and Ole Troan for their contributions to this work.

11. References

11.1. 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>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[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>.

[RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July 2015, <http://www.rfc-editor.org/info/rfc7599>.

11.2. Informative References

[I-D.boucadair-softwire-dslite-yang] Boucadair, M., Jacquenet, C., and S. Sivakumar, "YANG Data Model for the DS-Lite Address Family Transition Router (AFTR)", draft-boucadair-softwire-dslite-yang-04 (work in progress), June 2016.

[I-D.ietf-netmod-routing-cfg] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-22 (work in progress), July 2016.

[I-D.sivakumar-yang-nat] Sivakumar, S., Boucadair, M., and S. <>, "YANG Data Model for Network Address Translation (NAT)", draft-sivakumar-yang-nat-04 (work in progress), March 2016.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <http://www.rfc-editor.org/info/rfc6991>.

Authors' Addresses

Qi Sun
Tsinghua University
Beijing 100084
P.R. China

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

Hao Wang
Tsinghua University
Beijing 100084
P.R. China

Phone: +86-10-6278-5822
Email: wangh13@mails.tsinghua.edu.cn

Yong Cui
Tsinghua University
Beijing 100084
P.R. China

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

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

Email: ian.farrer@telekom.de

Sladjana Zoric
Deutsche Telekom AG
CTO-IPT,Landgrabenweg 151
Bonn, NRW 53227
Germany

Email: sladjana.zoric@telekom.de

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Rajiv Asati
Cisco Systems, Inc.
7025 Kit Creek Rd.
RTP, NC 27709
USA

Email: Rajiva@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: March 1, 2018

H. Xu
Q. Sun
China Telecom
Y. Fu
CNNIC

August 28, 2017

A Redundancy Mechanism for Dual-Stack Lite
draft-xu-v6ops-dslite-redundancy-01

Abstract

Dual-Stack Lite is a solution to offer both IPv4 and IPv6 connectivity to customers that are addressed only with an IPv6 prefix. This document provide a redundancy mechanism for Dual-Stack 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. The internet service provider no longer to provide publice IPv4 address but an IPv6 prefix to the customers as the issue of the IPv4 public address shortage. 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. As the exhaustion of the public IPv4 address, service providers have deployed DS-Lite in their network widely in nowadays, where a large number of customers are located. These customers within a network which is served by a single CGN function embedded in AFTR element may experience service degradation due to the presence of the single point of failure or loss of state information. Therefore, redundancy capabilities of the AFTR devices are strongly desired in order to deliver highly available services to customers. Failure detection and repair time should be therefore shortened.

This document describes a redundancy mechanism for DS-Lite. Some deployment consideration and recommendations for network elements are also provided.

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. Reliability Considerations of AFTR

As described in [RFC6908], for the robustness, reliability, and load distribution purposes, operators may deploy multiple AFTRs in their network. There are many deployment mechanism for the AFTR in ISP network, the most common type are distribution mode and centralization mode.

For the distribution mode, the CGN card is integrated into the free slot of the BRAS in a metro network. As the BRAS integrates the AFTR function of DS-lite, it provides DS-Lite connection service for a small area customers in this metro network. The service providers always integrated two CGN cards in the BRAS for reduncdancy consideration as the primary AFTR and backup AFTR.The capital cost of this mode is expensive because it always need two CGN cards for every BRAS. But 50 percent of these cards are idle most of time so that it is a big waste of money. There are various types and versions of BRAS have been deployed in the service provider’s network . Some of them have been used for over ten years and may not support the card insertion. Some of them may also don’t have free slot for the CGN card. It is not operational to replace all of them in a short period which result that it could deploy DS-Lite in some area and others can not in the same metro network.

For the centralization mode, a stand-alone AFTR device is deployed nearby the core router device at the exit of a metro network. It provides the DS-Lite connection service for the whole customers in this metro network. Service providers always deploy two stand-alone AFTR devices nearby the two core router device for the load distribution and redundancy purpose. The capital cost of this mode is more less than the distribution mode. It does not consume the slot resource of the BRAS. But it takes a big chanllenge for AFTR device for this mode in the large scale metro network because it takes performance requirements for the speed of the session creation and the maximum number of session maintenance. On the other side, it will create exta traffic when the users belong to the same BRAS are conmmunicating with eath other because it will connect to the AFTR device in the centralization mode first. It is a waste of bandwidth.

As described above, whether to use distribution mode or centralization mode depends on the trade-off between the investment and operational efficiency requirement of the service providers.

4. The Redundancy Machanism Overview

The fundamental principle of redundacncy machanism is to make the centralization mode to backup for the distribution mode. The architecture of the redundancy mechanism is illustrated as Figure 1. It deploys one AFTR card into every BRAS which surport card insertion in metro network, as to provide bassic distributed DS-Lite connection service. Moreover, it deploy two stand-alone AFTR device near the core router at the exit of the metro network. So it could provide the DS-lite connection service for the users of the BRAS which don’t surppport card insertion and don’t have free slot for the AFTR card. One advantage of this mechanism is that the stand-alone AFTR device is not only a redundancy device but also can provide DS-Lite conection service for the BRAS without AFTR card slot. Then the IGP routing would be configured on the BRAS which has the AFTR card insertion.

```
                         +-------+ +-------+
                         | IPv4  | | IPv6  |
                         | APP   | | APP   |
                         +---+---+ +----+--+
                             |    X     |
                             |   / \    |
                         +---+--+   +---+---+
                         |      |   |       |
                         | CR   +---+  CR   |
                         |(AFTR)|   | (AFTR)|
                         +---+--+   +----+--+
                             |   \ /     |
                             |    X      |
                         +---+---+  +----+--+
                         |       |  |       |
                         | BRAS  |  | BRAS  |
                         | (AFTR)|  |       |
                         |       |  |       |
                         +---+---+  +----+--+
                             |           |
                         +---+---+  +----+--+
                         |       |  |       |
                         |  B4   |  |  B4   |
                         |       |  |       |
                         +-------+  +-------+
```

Figure 1: The architecture of the redunancy mechanism

It is made that the routing prior selected to the AFTR card on the BRAS and then selected the AFTR stand-alone device near the core router through the Metric value configuration.As the metric values of the two stand-alone AFTR device in centralization mode are the same, it ensure that the traffic of the same session would be forwarded to the same centralized AFTR device by the random selection of the hash algorithm.This mechanism is based on the IPv6 anycast function: when the AFTR card in distribution mode is breakdown,the AFTR address in router advertise message will disappear in the IGP routing table.The IP address of AFTR device in centralization mode is becoming the optimal routing. All the traffic for DS-Lite will be directed to the AFTR decive in the centralization mode as to keep the application alive.

5. The difference between the software process of the BRAS

The software process of the BRAS for distribution mode is described as Figure 2

```
                  ----------
                ///          \\\
              //                \\
             |   The traffic for  |
            |  IPv6 is flow into   |
             |  the inbound card  |
              \\                //
                \\\          ///
                   -----+----
                        |
            +-----------+-----------+
            |  Decapsulation for    |
            |  PPP                  |
            +------------+----------+
                         |
            +------------+----------+
            |   Look for IPv6 FIB   |
            +-----------|-----------+
                     ---|-----
                  ///         \\\
               /                 \      +------------------+
              |  Whether the AFTR  |  Y  |                  |
             |   Card Breakdown?    -----| PPP Session      |
               \                 /      | Interruption     |
                  \\\         ///       |                  |
                     ---|-----          +-------|----------+
                      N |                       |
               +--------------------+   +------------------+
               |   The local AFTR   |   |                  |
               |   decapsulated     |   | The DS-Lite      |
               |   the packets from |   | Service are      |
               |   the IPv6 tunnel  |   | Terminated       |
               +--------|-----------+   +------------------+
                        |
               +--------|-----------+
               |      NAT44         |
               +--------|-----------+
                        |
               +--------|-----------+
               | Look for IPv4 FIB  |
               +---------|----------+
                      ---|-----
                   ///         \\\
                  /                \
                |   The traffic for |
               |    IPv4 flow out to |
               |    the outbound     |
                |   card            |
```

```
                  \               /
                   \\\         ///
                      ---------
```

     Figure 2: The software process of the BRAS for distribution mode

And the software process of the BRAS for the new machanism is described as Figure 3:

```
                  ----------
                ///          \\\
              //                \\
             |   The traffic for  |
            |  IPv6 is flow into   |
             |  the inbound card  |
              \\                //
                \\\          ///
                   -----+----
                        |
            +-----------+-----------+
            |  Decapsulation for    |
            |  PPP                  |
            +------------+----------+
                         |
            +------------+----------+
            |   Look for IPv6 FIB   |
            +-----------|-----------+
                        |
                     ---|-----
                  ///         \\\
               /                   \      +------------------+
              |  Whether the AFTR   |  Y  | The traffic for |
             |   Card Breakdown?     -----| IPv6 flow are   |
               \                   /      | out to the      |
                  \\\         ///         | outbound card   |
                     ---|-----            +------|----------+
                      N |                        |
               +--------------------+     +-------------------+
               |   The local AFTR   |     |                  |
               |   decapsulated     |     | The centralized  |
               |   the packets from |     | AFTR decapsulated|
               |   the IPv6 tunnel  |     | the packets from |
               +--------|-----------+     | the IPv6 tunnel  |
                        |                 |                  |
               +--------|-----------+     +------|-----------+
               |      NAT44         |            |
               +--------|-----------+     +------|-----------+
                        |                 |                  |
```

```
               +--------|----------+      |       NAT44      |
               | Look for IPv4 FIB  |      +-------|-----------+
               +---------|----------+             |
                         |                +-------|-----------+
                      ---|-----           |                   |
                    ///         \\\       | Look for IPv4 FIB|
                   /              \       +-------|-----------+
                  |   The traffic for |           |
                 |    IPv4 flow out to |          |
                 |    the outbound     |          |
                  |   card            |           |
                   \              /            /-|-------\
                    \\\         ///        ////          \\\\
                       ---------          |   The traffic for |
                                         |   IPv4 flow out to  |
                                          |    the outbound   |
                                           \\\\   card    ////
                                               \---------/
```

Figure 3: The software process of the BRAS for new mechanism

As compared between Figure 2 and Figure 3, the main difference for the new mechanism is that if the local AFTR card breakdown, the DS-Lite service can be maintained as the backup AFTR will take over the function to keep the application alive.

6. New requirements for the AFTR device

For this DS-Lite redundancy mechanism, there are some new requirements for the AFTR device as below:

1. If the ditribution AFTR card breakdown, the AFTR device SHOULD ensure that the traffic will not direct to the other distribution AFTR card.

2. It should use FQDN to decribe the AFTR in the DHCPv6 option as described in [RFC6334].

3. How many ditribution AFTR device could be covered by one centralization AFTR device will be different depends on the deployment by different ISPs.

4. The speed of the session creation for the centralized AFTR device could be calculated by a formula.

7. Security Considerations

The AFTR device of centralization mode will accept the tunnel request from the all DS-Lite users in the metro network. It needs aditional requirements to prevent from the spoofing attack.

1. Only the user passed the authentication could be assigned IPv6 prefix from the BRAS.

2. After assigned the IPv6 prefix to the authorized user, the BRAS will report this address to the AAA sever for recording.

3. Create a local database in the AFTR device of he centralized mode to record the IPv6 prefix of the authorized user.

4. Create an interface of the AAA sever for the AFTR device to synchrionize the IPv6 prefix of the authorized user between the AAA sever to the local database of the AFTR.

5. When the BRAS receive a new request for a new tunnel, it will compare with the source IPv6 prefix with the local database of the AFTR. If it is match, it will accept the request for tunnel. If not, it will ignore the request regarding it is from a illegal user and report the illegal address to the network management system.

6. If the authorized user offline, BRAS will ask the AAA server to delete this user from the database.

8. IANA Considerations

This draft does not request any IANA action.

9. Acknowledgements

The authors would like to thanks the valuable comments made by XXX and other members of v6ops WG.

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

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, <https://www.rfc-editor.org/info/rfc2119>.

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

[RFC6334] Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite", RFC 6334, DOI 10.17487/RFC6334, August 2011, <https://www.rfc-editor.org/info/rfc6334>.

10.2. Informative References

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

[RFC6908] Lee, Y., Maglione, R., Williams, C., Jacquenet, C., and M. Boucadair, "Deployment Considerations for Dual-Stack Lite", RFC 6908, DOI 10.17487/RFC6908, March 2013, <https://www.rfc-editor.org/info/rfc6908>.

Authors' Addresses

Honglei Xu
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing, 100035
P.R. China

Email: xuhl.bri@chinatelecom.cn

Qiong Sun
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
P.R. China

Email: sunqiong.bri@chinatelecom.cn

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

Email: fuyu@cnnic.cn