

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 11, 2011

M. Boucadair
C. Jacquenet
France Telecom
J. Song
Q. Niu
ZTE Corporation
October 8, 2010

Procedure to bypass DS-Lite AFTR
draft-boucadair-softwire-cgn-bypass-03.txt

Abstract

This document proposes a solution to avoid the use of two stateful DS-Lite AFTR devices when both end-points are located behind different AFTR devices. For this purpose a new IPv6 extension header, called Tunnel Endpoint Extension Header (TEEH), is defined. The proposed procedure encourages the use of IPv6 between DS-Lite AFTR nodes as a means to avoid the unnecessary crossing of AFTR devices. A Flow Label based solution is also described.

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on April 11, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the

document authors. All rights reserved.

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

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
1.1. Purpose	4
1.2. Terminology	4
1.3. Contribution of this Draft	5
2. Overall Scenarios	5
3. Tunnel Endpoint Extension Header	7
4. AFTR Bypass Procedure	8
4.1. Overview	8
4.2. Operational Mode	8
5. Flow Label Based Alternative	12
6. IANA Considerations	14
7. Security Considerations	14
8. Acknowledgements	15
9. References	15
9.1. Normative References	15
9.2. Informative References	15
Appendix A. Alternative Solution	16
Authors' Addresses	18

1. Introduction

1.1. Purpose

The main purpose of this document is to investigate solutions to avoid the solicitation of some of the (AFTR-embedded) NAT capabilities along the path between two hosts located behind AFTR devices.

The advantages of this procedure include:

- o Better one-way delay: No need to check the payload in the originating AFTR and no need to execute ALG operations twice;
- o Optimised routing path;
- o Better use of available AFTR resources;
- o Enhance robustness: an AFTR device is withdrawn from the data path. The stateful nature of DS-Lite AFTR devices will affect the overall performance of the communication. This performance may be even more affected when two AFTR devices need to be crossed to establish the communication.

1.2. Terminology

Within this memo, the term AFTR is used to refer to both following schemes:

- o an AFTR function embedded in a router, and/or
- o a standalone AFTR with limited routing capabilities (redirection capabilities to the AFTR are being enabled in an external router).

An outbound AFTR is referred to as Source AFTR.

An inbound AFTR is called a Target AFTR.

In the example illustrated in Figure 1, if we suppose that A initiates a communication towards B, AFTR1 is the Source AFTR and AFTR2 is the Target AFTR.

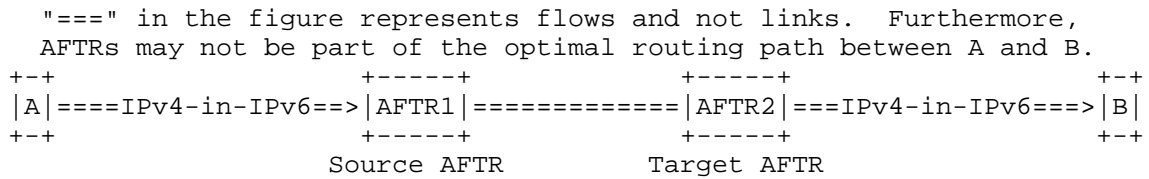


Figure 1: Source and Target AFTR

1.3. Contribution of this Draft

This document proposes a solution to avoid invoking NAT capabilities when several DS-Lite AFTR devices [I-D.ietf-softwire-dual-stack-lite] are involved in the data path. This document encourages the use of IPv6 for forwarding traffic between two AFTR devices.

This memo focuses primarily on the AFTR devices deployed in the same administrative domain. AFTRs located in distinct administrative domains are out of scope.

This document does not make any assumption on the services that may require the establishment of direct communications between hosts located behind AFTR devices. Examples of services would be P2P or hosting FTP/HTTP/SIP server behind a DS-Lite CPE.

In order to offload AFTR devices, application-specific solutions (e.g., [I-D.carpenter-behave-referral-object] [I-D.boucadair-mmusic-altc], [I-D.boucadair-dispatch-ipv6-atypes]) may be required to be implemented in order to prefer native IPv6 communications rather than crossing AFTR devices.

The implementation of the proposed procedure is not motivated in a context where the percentage of traffic involving two AFTR devices is minor (e.g., 1%). Nevertheless, as a side effect, Tunnel Endpoint Extension Header (TEEH) (Section 3) may be used to withdraw an AFTR from the data path, when both participants are managed by the same AFTR.

When TEEH is not supported, Two alternatives solutions are described in Section 5 and Appendix A.

2. Overall Scenarios

This section provides an overview of targeted scenarios.

Figure 2 illustrates the communication between two hosts that are located behind an AFTR device. Two NAT operations are required to be

performed for the establishment of successful communication between A and B. The stateful nature of a DS-Lite AFTR device will presumably affect the overall performance of the communication. This performance may be even more affected when two AFTR devices need to be crossed to establish the communication.

Prior to sending datagrams to B, A has retrieved the IPv4 public address of B owing to DNS resolution, third party referral, etc.

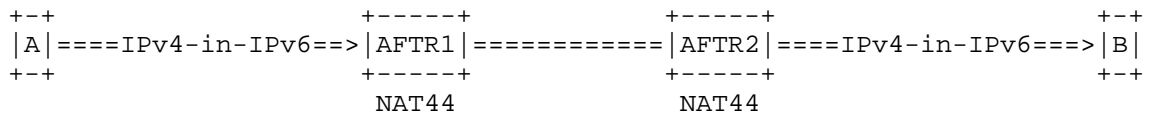


Figure 2: Nominal behaviour

A first optimisation scenario is shown in Figure 3 where NAT capabilities of the Source AFTR are not solicited. A second optimisation scenario is shown in Figure 4 where NAT capabilities of the Target AFTR are not solicited. The latter is not a valid scenario since the destination is seen with a public IPv4 address which is managed by the Target AFTR (consequently, a NAT44 state must be instantiated in the Target AFTR). The last configuration, illustrated in Figure 5, aims at avoiding the use of NAT capabilities in both Source and Target AFTRs. This configuration is impossible to implement since the remote destination must always be seen with an external public IPv4 address (and/or an IPv6 one). Having an external IPv4 address means that a AFTR has assigned an IPv4 address and port number for that host. Therefore, all the incoming IPv4 traffic must cross that AFTR.

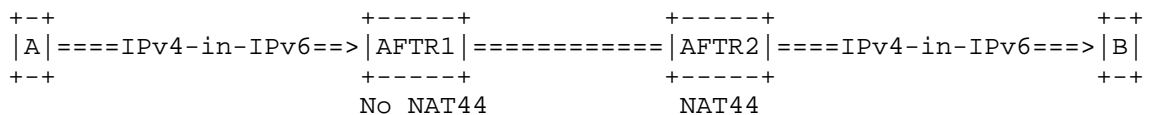


Figure 3: Avoid Source NAT44

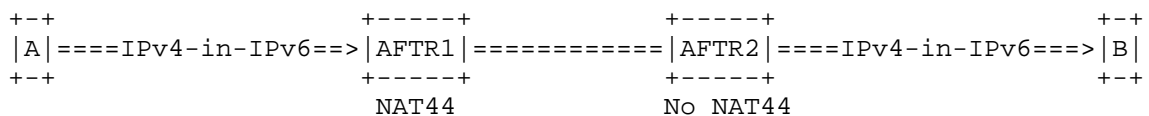


Figure 4: Avoid Target NAT44

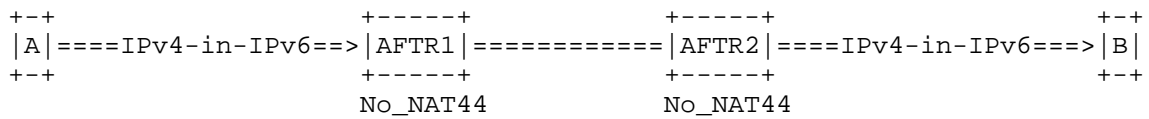
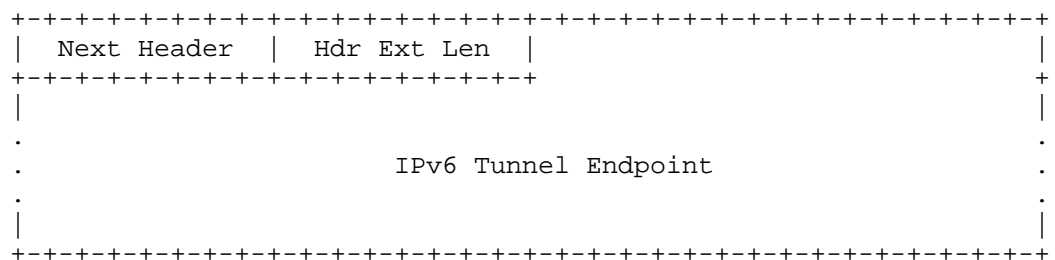


Figure 5: Avoid all NAT44

3. Tunnel Endpoint Extension Header

TEEH is a new IPv6 extension header which is used to inform the remote party about the destination IPv6 address to be used when issuing a response. Particularly, TEEH is used by the Source AFTR to inform the Target AFTR about the IPv6 address of a customer's device attached to the Source AFTR. Therefore, the Target AFTR acts as an inbound AFTR for that customer's device.

The format of the Tunnel Endpoint header is shown in Figure 6.



[NOTE: the format of TEEH may change in the next version of the document to include other information such as the scope for instance]

Figure 6: Tunnel Endpoint Extension Header

The description of the fields is as follows:

- o Next Header (8-bit): Identifies the type of header immediately following the TEEH header.
- o Hdr Ext Len (8-bit, unsigned integer): Length of the Tunnel Endpoint header in 8-octet units, not including the first 8 octets.
- o IPv6 Tunnel Endpoint: Encloses an IPv6 address that should be used as source of the encapsulated IPv4-in-IPv6 response. This field must be padded to ensure that the TEEH length is a multiple of 8 octets.

When TEEH is included in a received IPv4-in-IPv6 datagram, the answer SHOULD be sent to the IPv6 address conveyed in the TEEH.

When TEEH is inserted by a AFTR in an IPv4-in-IPv6 datagram sent to a customer's device, the IPv6 address included in the TEEH SHOULD be used as destination IPv6 address of subsequent IPv4-in-IPv6 messages.

4. AFTR Bypass Procedure

4.1. Overview

Each CPE (which embeds a B4 function) is notified of the IPv6 reachability information of (one of) the available DS-Lite AFTRs (e.g., using [I-D.ietf-softwire-ds-lite-tunnel-option]). In addition, the CPE must support at least one encapsulation scheme to convey privately-addressed IPv4 traffic into IPv6 datagrams. The CPE behaves as defined in [I-D.ietf-softwire-dual-stack-lite].

A dedicated IPv6 prefix (pref6_aftr) is used to convey the traffic between AFTR nodes.

The following configuration tasks should be undertaken:

- o Each AFTR is provided with an IPv4 address pool (IPv4@) for its NAT operations;
- o An IPv4-Converted IPv6 prefix [I-D.ietf-behave-address-format] is also assigned to each AFTR. This IPv6 prefix embeds the IPv4 net: pref6_aftr+IPv4@.
- o This IPv6 prefix is injected in a routing protocol (IGP/MP-BGP/i-BGP, or softwire full mesh is used between AFTRs). This route announcement is assumed to be performed by the AFTR itself or by the router which is responsible for redirecting the traffic to a AFTR. When pref6_aftr+IPv4@ is found on routing table, it is used as a "hint" to detect that the IPv4 address is provisioned on a AFTR device.

An operational mode to bypass an AFTR is described in Section 4.2.

4.2. Operational Mode

IPv4-in-IPv6 encapsulated datagrams issued by a CPE are received by an AFTR device (Step 1). This AFTR de-capsulates the datagram and retrieves the destination IPv4 address. Then, it proceeds to a route lookup to check whether a route towards "pref6_aftr+destination IPv4@" is installed. If not, it proceeds with traditional NAT

operations. Otherwise (i.e., a route is found. This means that the destination is located behind an AFTR), no NAT44 state is instantiated by the Source AFTR. The datagram is then encapsulated in IPv6 datagram with an IPv6 destination address equal to "pref6_aftr+destination IPv4 @::x" (refer to [I-D.ietf-behave-address-format] for more information on how to build IPv4-Converted IPv6 addresses).

As for the source IPv6 address of the encapsulated datagram, two schemes may be envisaged:

(1) Maintain the same source IPv6 address as per the datagram received from the customer's device. The deployment of this alternative requires the activation of security association to secure the exchange between the Source and Target AFTR. A trust relationship must be configured.

(2) A new extension header (called TEEH for Tunnel Endpoint Extension Header, defined in Figure 6) is inserted to indicate where to send the response back. The value of the extension header is an IPv6 address of the source CPE (as stored in the Source AFTR).

The datagram is forwarded to the next hop until being delivered to a Target AFTR (Step 2).

- If a NAT entry is instantiated on that AFTR, the datagram is processed. Additionally, the source IPv6 address of the received datagram or the content of the TEEH is stored by the AFTR. This information will be used to send back the response.

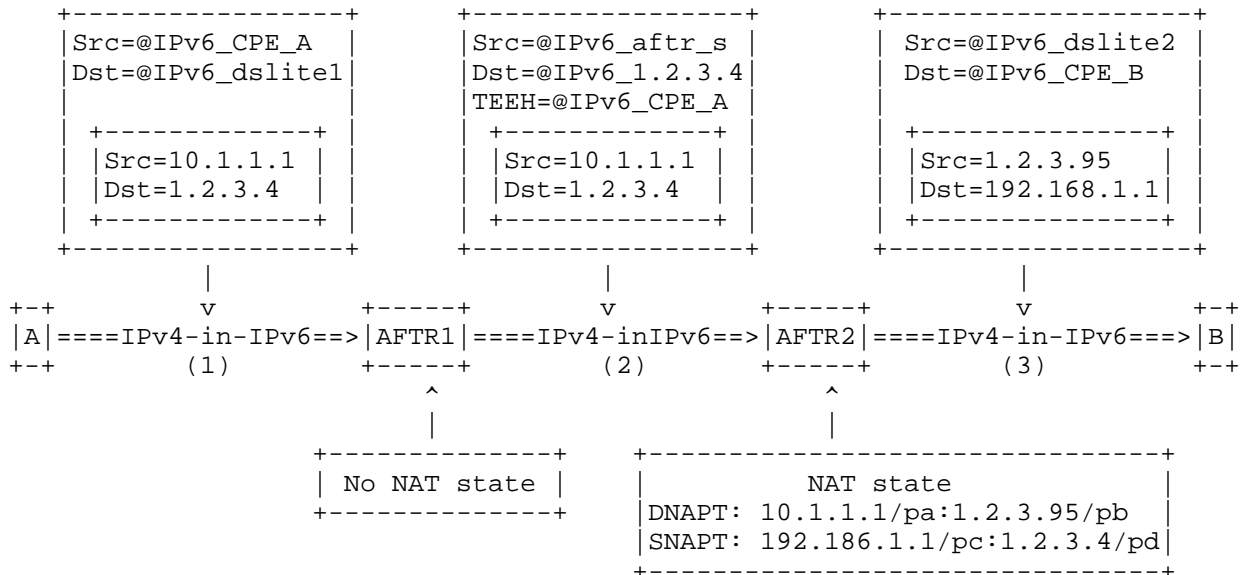
In addition to re-writing destination IPv4 address+port (i.e., DNAPT for Destination NAPT), the IPv4 source address and the port number are also modified (referred to as SNAPT for Source NAPT). The translation of the source IPv4 address is required to avoid overlapping private IPv4 addressing in the destination home realm. A public IPv4 address belonging to the Target AFTR pool is used to enforce SNAPT. This SNAPT operation does not alter the number of sessions that may be maintained by a given AFTR.

The resulting IPv4 datagram is then encapsulated in IPv6 and forwarded to its final destination (i.e., B in Figure 7) (Step 3).

An AFTR must be configured to accept TEEH only when it is issued by other AFTR devices. A filtering rule based on the source IPv6 address MAY be configured.

- Otherwise, the datagram is rejected/dropped/silently discarded.

Figure 7 illustrates the occurred flow exchanges.



pa, pb, pc and pd are port numbers. Only an excerpt of the NAT table is shown, IPv6 addresses are also maintained in the NAT table.

Figure 7: Outbound traffic

As for the response, B encapsulates IPv4 traffic in IPv6 datagrams that are forwarded to the AFTR as illustrated in Figure 8 and Figure 9 (Step 4). The AFTR then proceeds to NAT operations (both DNAPT and SNAPT). The resulting IPv4 traffic is then encapsulated in IPv6 and corresponding IPv6 datagrams are then forwarded to the IPv6 address of the remote destination as maintained in the NAT tables (Step 5). TEEH may be inserted to indicate the destination IPv6 address to be used for the subsequent messages (see Figure 8). Figure 9 shows the exchanged flows when TEEH is not used.

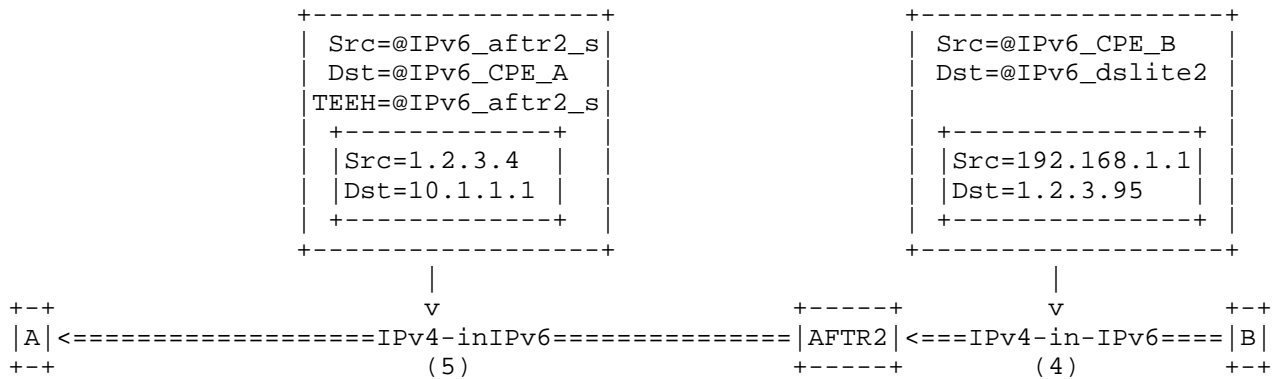


Figure 8: Incoming traffic with Option Header

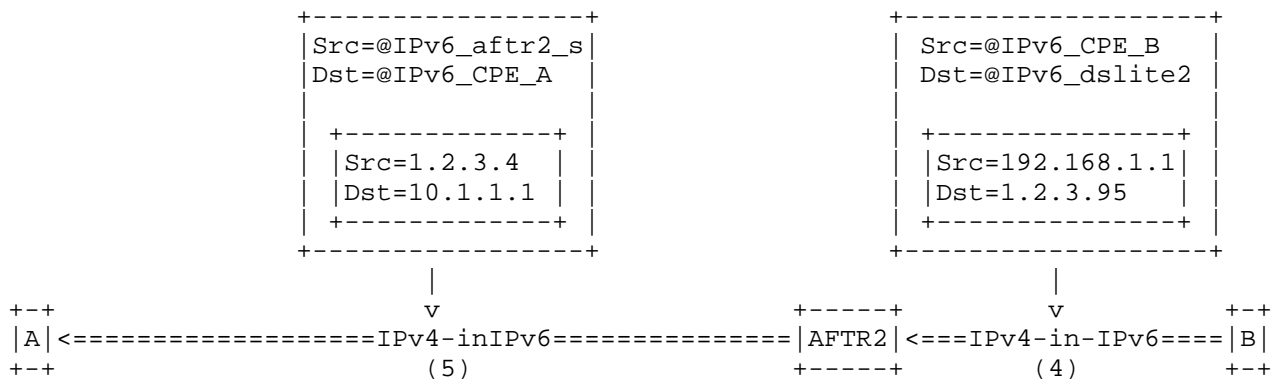


Figure 9: Incoming traffic without Option Header

For the remaining exchanges, either A uses the IPv6 address of AFTR2 to send subsequent messages owing to the presence of TEEH option (see Figure 8. The experienced behaviour is illustrated in Figure 10) or it uses the default behavior and it sends all IPv4 traffic to its attached AFTR1 (as illustrated in Figure 7).

A CPE must be configured to accept incoming IPv4-in-IPv6 traffic with a source address belonging to an IPv6 prefix used to address AFTR devices.

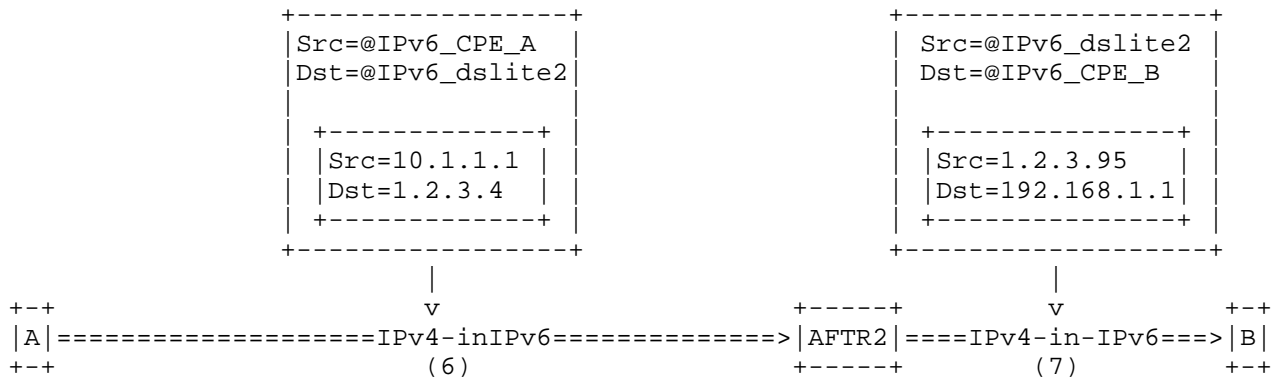


Figure 10: Withdraw Source CGN

As a result, NAT operations are enforced in one AFTR instead of two nodes. One AFTR is withdrawn from the path.

5. Flow Label Based Alternative

This alternative aims at avoiding two NAT operations without withdrawing an AFTR from the path and without adding a new IPv6 extension header.

Outbound flow exchanges are illustrated in Figure 11. Inbound flow exchanges are shown in Figure 12.

IPv6 is used to convey traffic between AFTR nodes. IPv4-Converted IPv6 addresses are used to detect whether the destination is also managed by an AFTR. No NAT state is then instantiated in the Source AFTR. Two AFTRs are maintained in the path but only one AFTR maintains a NAT state.

AFTR assigns a sequence number (or index) for every software between the AFTR and CPE. Sequence numbers must be generated by an AFTR to uniquely identify a given software.

The source AFTR sends the sequence number filled in flow label field of the IPv6 header to the target AFTR for indicating where to send the response back.

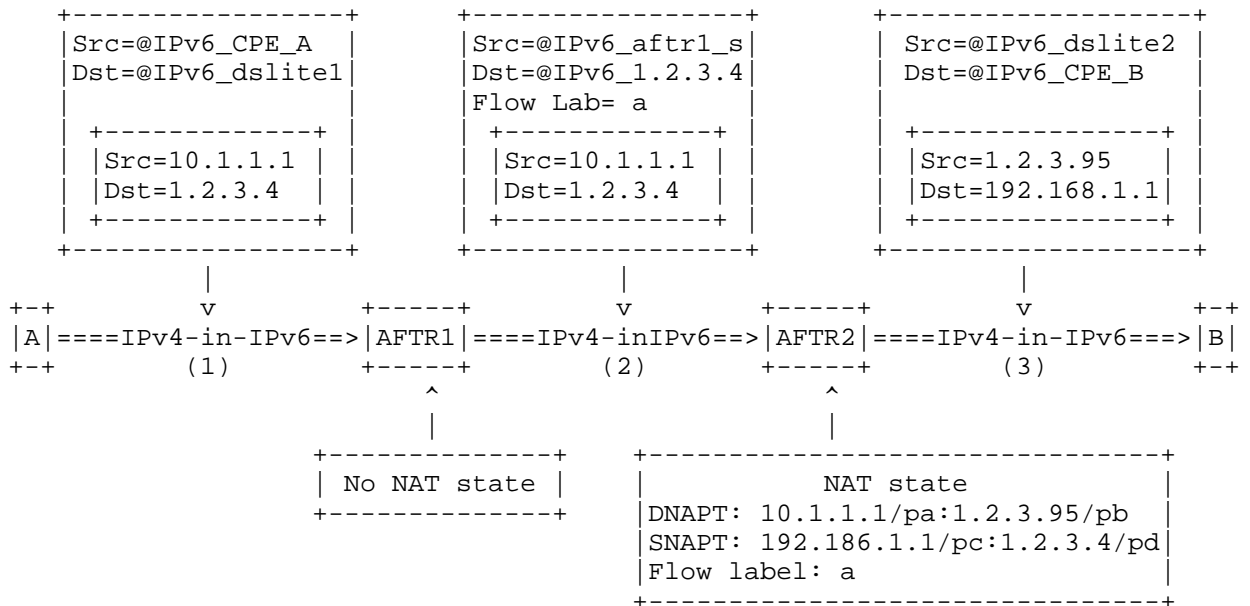


Figure 11: Outbound traffic

These steps are followed:

- o Step 1: A encapsulates its IPv4 datagram in IPv6 one and forwards the encapsulated IPv4-in-IPv6 datagram to its outbound AFTR.
- o Step 2: Once that datagram is received by AFTR1, it de- capsulates it and retrieves the IPv4 datagram. Moreover, the destination IPv4 address is returned. AFTR1 proceeds to a routing look up to check whether a route to pref6_aftr+destination IPv4@ is installed. If the answer is positive (i.e., the destination is managed by an AFTR), AFTR1 does not proceed to any NAT44 operation. The IPv4 datagram is then encapsulated in an IPv6 one and forwarded to AFTR2 (destination IPv6 address of the encapsulated datagram is pref6_aftr+IPv4@). The sequence number a of software between AFTR1 and A is filled in the Flow Label field of the IPv6 packet.
- o Step 3: AFTR2 receives that datagram. It de-capsulates the received datagram and retrieves the enclosed IPv4 one. AFTR2 checks if a NAT state is already instantiated towards the destination IPv4 address/port number. If the answer is positive, then it proceeds to DNAPT and SNAPT. AFTR2 keeps the sequence number a in the NAT table. The resulting datagram is then

forwarded to the IPv6 address of B (stored in AFTR2).

- o Step 4: B replies as per DS-Lite specification. o
- o Step 5: AFTR2 de-capsulates the received datagram and proceeds to DNAPT and SNAPT. The resulting IPv4 datagram is then encapsulated in an IPv6 one and the sequence number a is filled in the Flow Label field. The IPv6 packet is forwarded to AFTR1. o
- o Step 6: AFTR1 finds the softwire according sequence number a carried in the Flow Label field, then it forwards the packet to A.

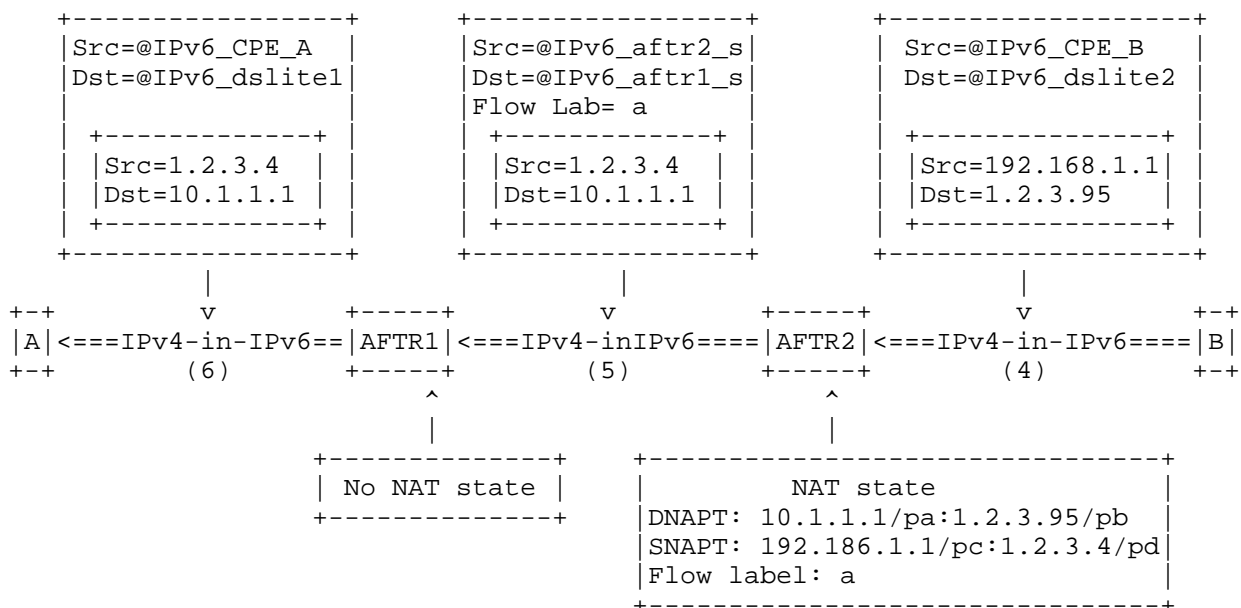


Figure 12: Inbound traffic

6. IANA Considerations

TBC.

7. Security Considerations

B4 element MUST be configured to accept incoming IPv4-in-IPv6 datagrams not issued by its outbound AFTR. All deployed AFTRs SHOULD share a security association to secure the use of the TEEH option.

8. Acknowledgements

The author would like to thank P. Levis, M. Kassi Lahlou, E. Burgey and D. Binet for their feedback and comments.

9. References

9.1. Normative References

- [I-D.ietf-behave-address-format]
 Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", draft-ietf-behave-address-format-10 (work in progress), August 2010.
- [I-D.ietf-softwire-dual-stack-lite]
 Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-06 (work in progress), August 2010.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

- [I-D.boucadair-dispatch-ipv6-atypes]
 Boucadair, M., Noisette, Y., and A. Allen, "The atypes media feature tag for Session Initiation Protocol (SIP)", draft-boucadair-dispatch-ipv6-atypes-00 (work in progress), July 2009.
- [I-D.boucadair-mmusic-altc]
 Boucadair, M., Kaplan, H., Gilman, R., and S. Veikkolainen, "Session Description Protocol (SDP) Alternate Connectivity (ALTC) Attribute", draft-boucadair-mmusic-altc-01 (work in progress), September 2010.
- [I-D.carpenter-behave-referral-object]
 Carpenter, B., Boucadair, M., Halpern, J., Jiang, S., and K. Moore, "A Generic Referral Object for Internet Entities", draft-carpenter-behave-referral-object-01 (work in progress), October 2009.
- [I-D.ietf-softwire-ds-lite-tunnel-option]
 Hankins, D. and T. Mrugalski, "Dynamic Host Configuration

Protocol for IPv6 (DHCPv6) Option for Dual- Stack Lite",
draft-ietf-softwire-ds-lite-tunnel-option-05 (work in
progress), September 2010.

Appendix A. Alternative Solution

This alternative aims at avoiding two NAT operations without withdrawing a AFTR from the path.

Outbound flow exchanges are illustrated in Figure 13. Inbound flow exchanges are shown in Figure 14.

IPv6 is used to convey traffic between AFTR nodes. IPv4-Converted IPv6 addresses are used to detect whether the destination is also managed by an AFTR. No NAT state is then instantiated in the Source AFTR. Two AFTR are maintained in the path but only one AFTR maintains a NAT state.

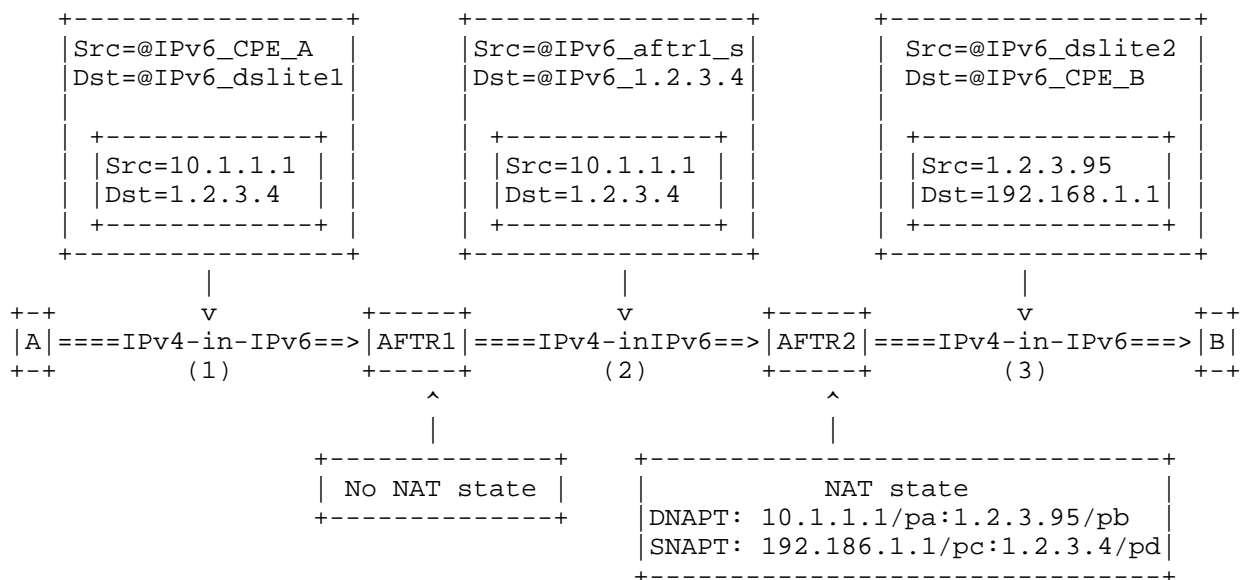


Figure 13: Outbound traffic

The following steps are followed

- o Step 1: A encapsulates it IPv4 datagram in IPv6 one and forwards the encapsulated IPv4-in-IPv6 datagram to its outbound AFTR. The

IPv6 address/FQDN of its outbound AFTR is provisioned using DHCP for instance.

- o Step 2: Once that datagram is received by the AFTR1, its de-capsulates it and retrieves the IPv4 datagram. Moreover, the destination IPv4 address is returned. AFTR1 proceeds to a routing look up to check whether a route to pref6_aftr+destination IPv4@ is installed. If the answer is positive (i.e., the destination is managed by an AFTR), AFTR1 does not proceed to any NAT44 operation. The IPv4 datagram is then encapsulated in an IPv6 one and forwarded to AFTR2 (destination IPv6 address of the encapsulated datagram is pref6_aftr+IPv4@). The source IPv6 address used by AFTR1 must identify unambiguously A.
- o Step 3: AFTR2 receives that datagrams. It de-capsulates the received datagram and retrieves the enclosed IPv4 one. AFTR2 checks if a NAT state is already instantiated towards the destination IPv4 address/port number. If the answer is positive, then it proceeds to DNAPT and SNAPT. The resulting datagram is then forwards to the IPv6 address of B (stored in AFTR2).
- o Step 4: B replies as per DS-Lite specifications.
- o Step 5: AFTR2 de-capsulates the received datagrams and proceeds to DNAPT and SNAPT. The resulting IPv4 datagram is then encapsulated in an IPv6 one and forwarded to AFTR1.
- o Step 6: AFTR1 checks its swapping states and forwards the packet to A.

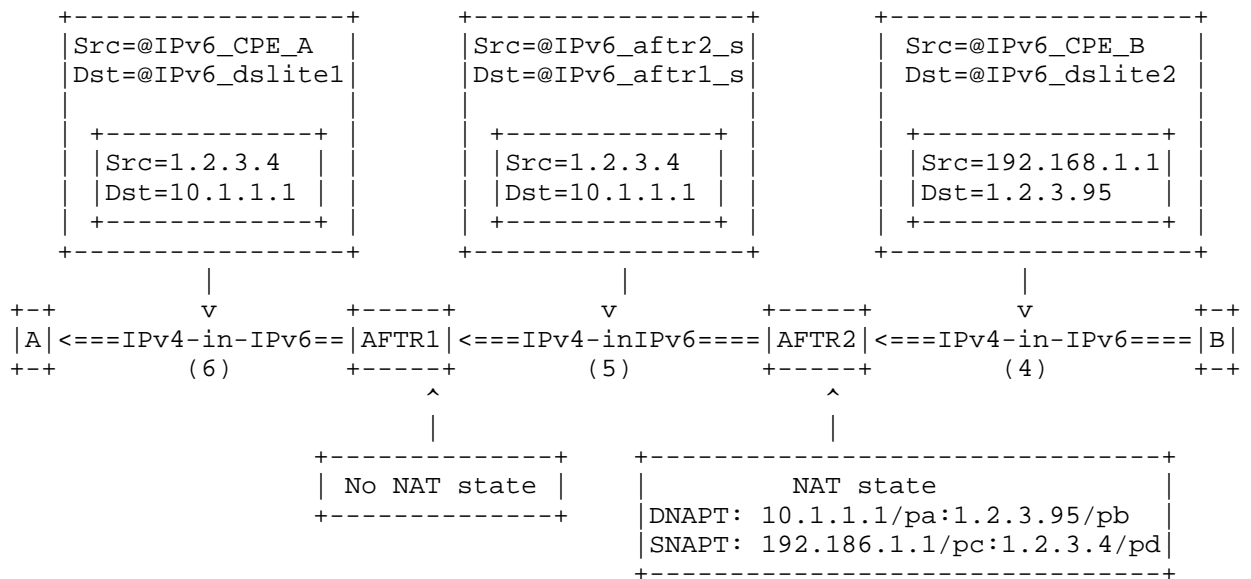


Figure 14: Inbound traffic

Authors' Addresses

Mohamed Boucadair
 France Telecom
 Rennes 35000
 France

Email: mohamed.boucadair@orange-ftgroup.com

Christian Jacquenet
 France Telecom
 Rennes 35000
 France

Email: christian.jacquenet@orange-ftgroup.com

Jun Song
ZTE Corporation
No.68,Zijinghua Road, Yuhuatai District
Nanjing, Jiangsu Province
China

Email: song.jun@zte.com.cn

Qibo Niu
ZTE Corporation
No.68,Zijinghua Road, Yuhuatai District
Nanjing, Jiangsu Province
China

Email: niu.qibo@zte.com.cn

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 8, 2011

M. Boucadair, Ed.
C. Jacquenet
JL. Grimault
M. Kassi-Lahlou
P. Levis
France Telecom
D. Cheng, Ed.
Huawei Technologies Co., Ltd.
Y. Lee, Ed.
Comcast
October 5, 2010

Deploying Dual-Stack Lite in IPv6 Network
draft-boucadair-softwire-dslite-v6only-00

Abstract

Dual-Stack lite requires that the AFTR must have IPv4 connectivity. This forbids a service provider who wants to deploy AFTR in an IPv6-only network. This memo proposes an extension to implement a stateless IPv4-in-IPv6 encapsulation in the AFTR so that AFTR can be deployed in an IPv6-only network.

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on April 8, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

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

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
1.1. General	4
1.2. Requirements	4
1.3. Overview	4
2. Terminology	6
3. Addressing	7
4. DS-Lite AFTR	8
4.1. Provisioning	8
4.2. Procedure	8
4.2.1. Processing Ingress Traffic from Customer Interface	9
4.2.2. Processing Ingress Traffic from Core Interface	9
4.3. Flows Examples	10
5. IPv6-IPv4 Interconnection Function (ICXF)	11
5.1. Provisioning	11
5.2. Procedure	11
6. Routing Architecture and Considerations	12
6.1. Static Routing	12
6.2. Dynamic Routing	12
7. Multicast Considerations	14
8. Fragmentation	14
9. Conclusions	14
10. IANA Considerations	15
11. Security Considerations	15
12. Acknowledgements	15
13. References	15
13.1. Normative References	15
13.2. Informative References	16
Appendix A. Changes Since 02	16
Authors' Addresses	16

1. Introduction

1.1. General

Dual-Stack lite (DS-lite) contains two major concepts: (1) Transport IPv4 packets over an IPv6 access network, and (2) Share a public IPv4 address to multiple users.

The B4 element resided in the customer premises is provisioned an global routable IPv6 address. It also establishes an IPv4-in-IPv6 tunnel to AFTR element. The hosts behind B4 elements are assigned with [RFC1918] addresses. When the B4 receives IPv4 datagram from it managed host, it will send the datagram over the IPv4-in-IPv6 tunnel to the AFTR.

AFTR element provides the NAT function and is responsible for sharing public IPv4 addresses to multiple B4 elements. It also requires direct IPv4 connectivity to send and received the NAT-ed datagram to the IPv4 network.

This model puts a demarcation in the network where the access network between B4 and AFTR can be IPv6-only and the network north of AFTR must be IPv4. Consider a service provider wants to extend the IPv6-only network boundary from the access network to the border of the network, this will force the AFTR to be deployed in the border and further away from the B4s. This memo describes a framework to allow a service provider to extend the IPv6-only network while to allow the AFTR to stay close to the B4s.

1.2. Requirements

- o [REQ1] Extend the IPv6-only boundary to the border of the network.
- o [REQ2] Only the AS Border Router has IPv4 connectivity.
- o [REQ3] The service provider provisions only IPv6 addresses to the customers but continues to provide IPv4 services to them.
- o [REQ4] The AFTR has only IPv6-connectivity and must be able to send and receive IPv4 packets.

1.3. Overview

DS-Lite [I-D.ietf-software-dual-stack-lite] directly connects users to IPv6 networks but at the same time provides IPv4 services by tunneling IPv4 packets over an IPv6 network.

AFTR element is the combination of an IPv4-in-IPv6 tunnel end-point

and an IPv4-IPv4 NAT implemented in the same node. In addition, the specification assumes that an AFTR is directly connected to the IPv4 network.

In some deployments where the service provider wants to deploy AFTR in the IPv6 core network. AFTR nodes may not have direct IPv4 connectivity. In this scenario, IPv4 packets after NAT44 function applied on an AFTR node need to be transported over the IPv6 core network to the IPv4 network. This memo proposes a framework for this scenario as an extension to the DS-Lite specification.

In this specification, we define a new stateless IPv6-IPv4 interconnection function (referred to as IPv6-IPv4 ICXF), in a border node located at the boundaries between the IPv6 and IPv4 networks. The AFTR discovers the ICXF address, and sends IPv4 encapsulated IPv6 packets after NAT44 function.

The ICXF may be hosted in an ASBR (Autonomous System Border Router) or a dedicated node located at the interconnection between IPv6 and IPv4 domains. A router that hosts the ICXF is referred to as an ICXF router.

When the AFTR receives a customer's outbound packet from B4 element, it de-capsulate the packet and perform standard NAT44 function. Since an AFTR does not directly connect to the IPv4 network, AFTR will encapsulate the NAT-ed IPv4 packet in an IPv4-in-IPv6 packet, with an IPv4-Embedded IPv6 destination address [I-D.ietf-behave-address-format], and forward it to an ICXF router located with direct connection to the IPv4 network. When the ICXF router receives the IPv4-Embedded IPv6 packet, it will de-capsulate the packet and forward the IPv4 packet based on the IPv4 destination address.

For an inbound IPv4 packet to B4 element, the ICXF router will encapsulate the IPv4 packet into IPv6 packet with the IPv4-Embedded IPv6 address and forward it to the appropriate DS-Lite AFTR node, which de-capsulates the IPv4 packet and then follows the normal procedure defined by DS-Lite architecture as if the IPv4 packet is received from a directly connected IPv4 network.

Figure 1 provides an overview of the global architecture. Customers are connected to the service domain via a CPE device. Several DS-Lite AFTR nodes are deployed to manage the traffic sent and received by the end-user terminal devices. The service domain is IPv6 only and interconnection with adjacent IPv4 realms is implemented using IPv6-IPv4 ICXF. The distributed deployment mode of AFTR nodes is motivated by several reasons such as optimizing intra-domain paths, avoiding single point of failure, minimizing the impact on geo-

localization services, minimizing the amount of customers to be impacted by an AFTR node failure, etc. AFTR deployment model varies from service provider to service provider and it is out of scope of this specification.

Note in this architecture, the DS-Lite B4 element (located in a CPE) and AFTR still behave exactly as defined in [I-D.ietf-softwire-dual-stack-lite], but with additional functions added to the AFTR when it does not directly connect to the IPv4 network. A new ICXF function is introduced to perform stateless IPv6-IPv4 interconnection. This specification defines new requirements on addressing scheme and routing. More details are provided in the following sections.

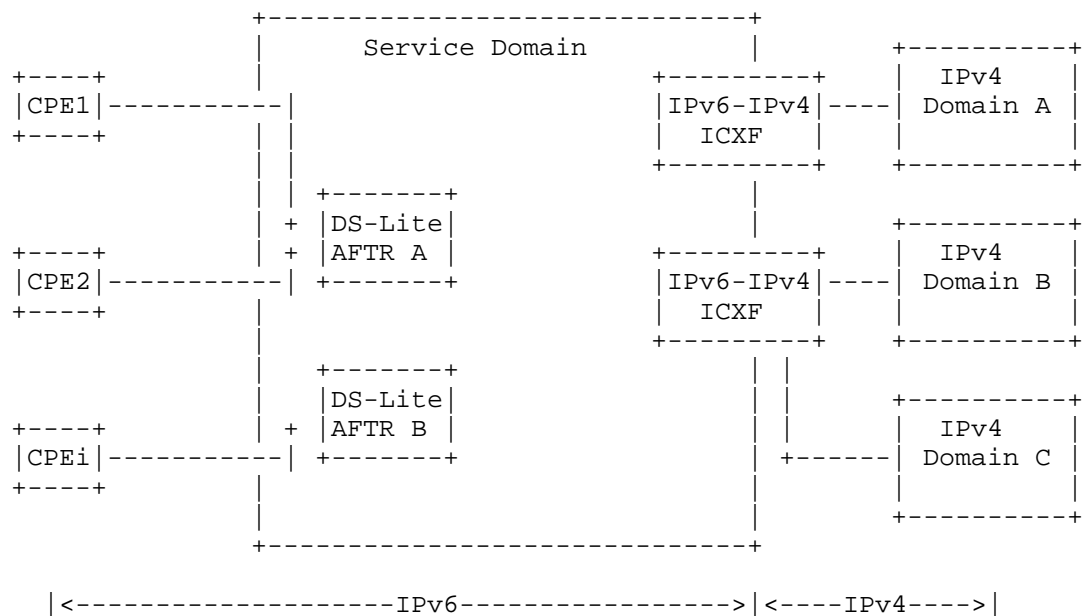


Figure 1: Architecture Overview

2. Terminology

This memo defines the following terms:

- o IPv6-IPv4 Interconnection Function (IPv6-IPv4 ICXF): refers to the function that de-capsulates (resp., encapsulates) the IPv6 (resp.,

IPv4) packet from DS-Lite AFTR node(s) and forwards the IPv4 (resp., IPv6) packets to the IPv4 (resp., IPv6) networks.

- o ICXF router: refers to the border router implemented with IPv6-IPv4 ICXF.
- o DS-Lite AFTR node: refers to the AFTR node whose behavior is specified in [I-D.ietf-softwire-dual-stack-lite]. In addition, this specification assumes that the DS-Lite AFTR node is only connected to an IPv6 network. The AFTR will encapsulate the IPv4 packet in an IPv6 packet (IPv4-in-IPv6) after the NAT44 function. The encapsulated IPv6 packet will be forwarded to the ICXF router. This IPv4-inIPv6 encapsulation is stateless.
- o Access segment: This segment encompasses both the IP access to the customers and to the service provider's network.
- o Interconnection segment: Includes all nodes and resources which are deployed at the border of a given AS (Autonomous System) a la BGP.
- o Core segment: Denotes a set of IP networking capabilities and resources which are located between the interconnection and the access segments.
- o Pref6: An IPv6 prefix assigned by LIR. This prefix is configured on both ICXF and AFTR.
- o FROM-AFTR Address: An IPv4-Embedded IPv6 address [I-D.ietf-behave-address-format] that combines an IPv6 prefix Pref6 and the destination IPv4 address.
- o TO-AFTR Address: An IPv4-Embedded IPv6 address [I-D.ietf-behave-address-format] that combines an IPv6 prefix Pref6 and a destination IPv4 address which configured in an AFTR NAT pool.

3. Addressing

For outbound IPv4 packets, the AFTR performs encapsulation and the ICXF router performs de-capsulation. For inbound IPv4 packets, the ICXF router performs IPv4-in-IPv6 encapsulation and an AFTR performs de-capsulation.

When an AFTR forwards an IPv6 packet with an IPv4 payload to an ICXF router, the source IPv6 address is one of the AFTR's IPv6 address, which is normally a global IPv6 address configured on an interface of

the node (e.g., an address of a loopback interface), and the destination IPv6 address is the FROM-AFTR Address.

When an ICXF router receives an IPv4 packet, it encapsulates the IPv4 packet with an IPv6 header where the source IPv6 address is the ICXF router's global IPv6 address and the destination IPv6 address is the TO-AFTR address. The TO-AFTR address is constructed by combining the Pref6 and the destination IPv4 address in the IPv4 packet. The destination IPv4 address is one of the addresses configured in the AFTR NAT pool.

In both cases, the Pref6 is an IPv6 prefix assigned by the service provider, and is used to construct an IPv4-Embedded IPv6 address. Figure 2 gives an example of the address format.

```
2a01:c::11000001001100111001000111001110 = 2a01:cc13:391c:e0::/56
|Pref6 | <-----193.51.145.206----->
```

Figure 2: Example for an IPv4-Embedded IPv6 Prefix

In this example, Pref6 is 2a01:c::/20 and the IPv4_Addr is 193.51.145.206. Then, the corresponding IPv6 prefix is: 2a01:cc13:391c:e0::/56. We use a /20 prefix for Pref6. However, an operator can decide to use any prefix length.

4. DS-Lite AFTR

4.1. Provisioning

The AFTR must be provisioned with a set of global IPv4 prefixes for NAT44 operations. In addition, an IPv6 prefix (i.e., Pref6) is configured in the AFTR. The Pref6 is used to construct FROM-AFTR addresses. The FROM-AFTR addresses are used in the destination address field of the IPv6 header for the IPv4-in-IPv6 packets.

4.2. Procedure

Figure 3 shows the input and output of a DS-Lite AFTR node.

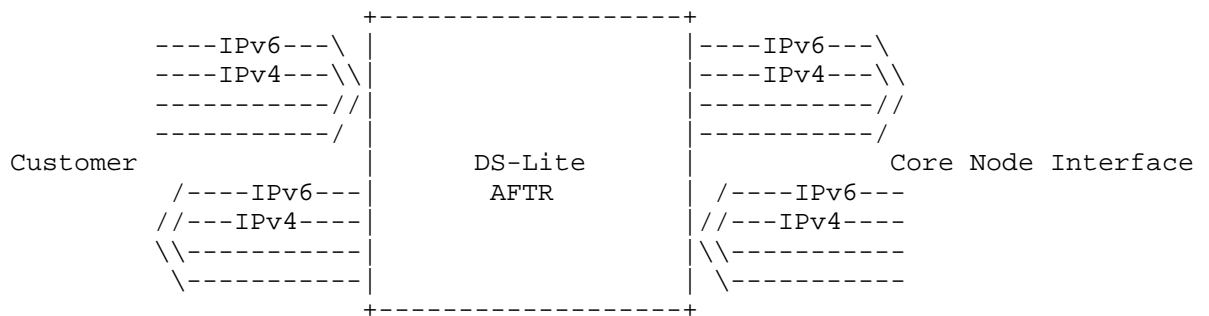


Figure 3: Modified DS-Lite AFTR

Two main (logical) interfaces may be distinguished in a DS-Lite AFTR node as follows:

- a. Interface with the customer device, i.e.- DS-Lite interface per [I-D.ietf-softwire-dual-stack-lite].
- b. Interface with core nodes. Note that the DS-Lite AFTR does not directly connect to an IPv4 domain.

4.2.1. Processing Ingress Traffic from Customer Interface

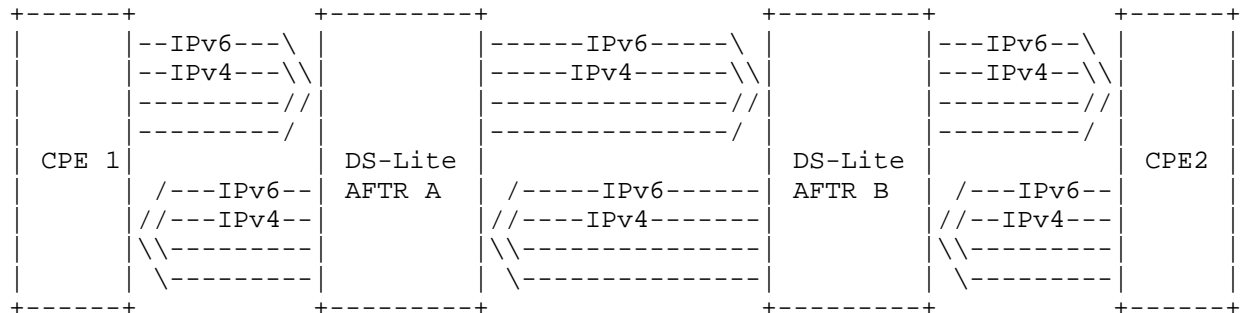
1. De-capsulate the IPv6 header from the IPv4-in-IPv6 packets (sent by the customer device) 2.
2. NAT the IPv4 packet and create an entry in the NAT binding table
3. Encapsulate the NAT-ed IPv4 packets in IPv6 with a destination IPv6 address built according to the addressing scheme described in Section 3. Encapsulated packet is forwarded based on the FROM-AFTR IPv6 address by standard routing. Depending on the target IPv4 address, the destination can be an AFTR node inside the service provider's domain if the IPv4 address is one of the addresses owned by another AFTR (See Figure 4). Or, the destination can be an ICXF router if the IPv4 address is external to the service provider.

4.2.2. Processing Ingress Traffic from Core Interface

1. De-capsulate the IPv6 header and extract the IPv4 packet.
2. Process the embedded IPv4 packet according to [I-D.ietf-softwire-dual-stack-lite].

3. Forward the resulting IPv6 packet to the corresponding B4.

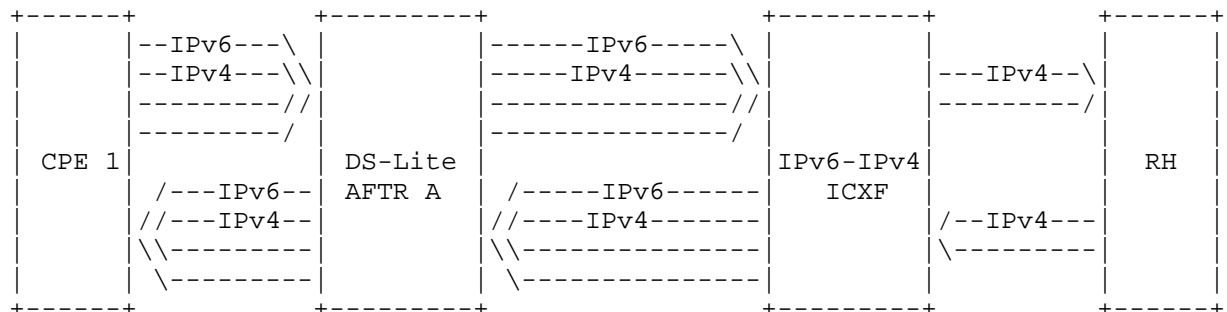
4.3. Flows Examples



Note that hosts connected to each CPE are not represented in the figure.

Figure 4: Flow Example involving two devices attached to distinct AFTRs

The following figure illustrates an example of CPE connected to a DS-Lite AFTR node, which establishes a communication with a remote host (referred to as RH) which is on an IPv4 network.



Note that host connected to CPE1 are not represented in the figure.

Figure 5: Flow Example involving only one device attached to a DS-lite enabled domain

5. IPv6-IPv4 Interconnection Function (ICXF)

ICXF is a border element that encapsulate IPv4 packets from external IPv4 network to AFTR and de-capsulate IPv6 packets from AFTR to external IPv4 network

Externally, the ICXF is connected to IPv4 network. It is an IPv4 router and performs standard IPv4 routing. It contains an IPv4 routing table and exchanges IPv4 prefixes to the internal and external peers.

Internally, the ICXF is connected to an IPv6 network and exchanges IPv4 prefixes to the AFTRs. Section 6 discusses the internal routing in details.

5.1. Provisioning

An IPv6-IPv4 ICXF router is provisioned with an IPv6 prefix (i.e., Pref6). Pref6 is used to build TO-AFTR addresses.

5.2. Procedure

Figure 6 shows the input and output of an IPv6-IPv4 ICXF.

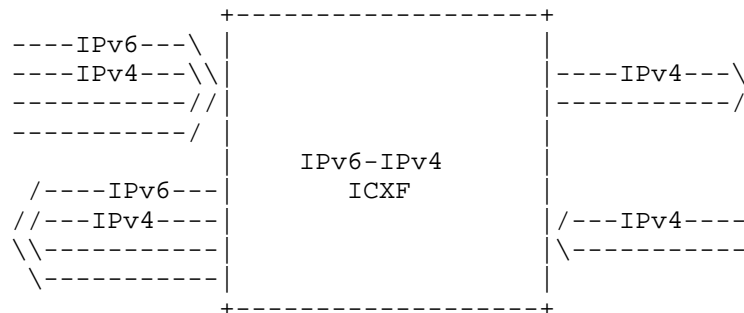


Figure 6: IPv6-IPv4 Interworking Function

When the ICXF router receives an IPv4 packet from an external IPv4 domain, it encapsulates the IPv4 packet in IPv6 packet using the following information:

- o Source IPv6 address: One of its own IPv6 addresses.
- o Destination IPv6 address: TO-AFTR Address which is an IPv4-Embedded IPv6 address using the Pref6 and destination IPv4 address

of the encapsulated IPv4 packet.

As for the outbound IPv6 packets, an ICXF router performs de-capsulation and forwards the embedded IPv4 packets to the connected IPv4 networks according to IPv4 routing rule.

6. Routing Architecture and Considerations

This section describes the routing consideration to support this specification, i.e.- how an IPv6 packet with an IPv4-Embedded IPv6 destination address is forwarded from a DS-Lite AFTR to an ICXF router, and vice versa, in the IPv6 network.

When a DS-Lite AFTR forwards IPv4-in-IPv6 packets to an ICXF router, the destination IPv6 address is an IPv4-Embedded IPv6 address, where the Pref6 is an IPv6 prefix assigned to the service provider and the IPv4 address is reachable through one or more ICXF routers. The forwarding decision can be made based on static or dynamic routing.

6.1. Static Routing

The AFTR is configured with static routes, and each static route points to an IPv4-Embedded IPv6 prefix. Alternatively, the AFTR can contain a default route where the default is the ICXF.

6.2. Dynamic Routing

Dynamic routing is more desirable for the deployments where there are multiple DS-Lite AFTRs and ICXF routers. This specification suggests four dynamic routing options as documented below:

Option 1:

- o AFTRs and ICXF routers are configured as a Softwire Mesh [RFC5565] and iBGP is used to exchange IPv4 reachability information. AFTR and ICXF will peer with each other over iBGP and exchange their IPv4 reachability. Each AFTR and ICXF will compute an IPv4 routing table based upon the BGP table. Given an IPv4 network managed by the AFTR or ICXF, the next-hop of this network is the IPv6 address of the AFTR or ICXF.
- o Pros: This routing option offers an optimized forwarding for IPv4-in-IPv6 packets in the IPv6 network.
- o Cons: DS-Lite AFTRs and ICXF routers must peer in iBGP and storing BGP routes on all these nodes.

Option 2:

- o ICXF router advertises its IPv4 reachability information in IGP. This routing option does not require AFTRs and ICXFs to be iBGP peers. For the AFTR IPv6 routing table, it contains all FROM_AFTR prefixes and the ICXF IPv4 reachability information in the form on IPv4-Embedded IPv6 prefixes (i.e., Pref6 + ICXF IPv4 routing information).
- o Pros: Given that the ICXF advertises all its IPv4 network reachability in IPv6 network, the AFTR can choose the best path to forward the packet.
- o Cons: This optimization has a drawback: ICXF routers are required to advertise its full IPv4 reachability to in IGP. As such, IPv6 routers will maintain the full IPv4 reachability in its IPv6 routing table.

Option 3:

- o With this option, each ICXF router advertises a Pref6 (Section 5.1) in the IPv6 IGP. An AFTR forwards an IPv4-in-IPv6 packet always to a nearest ICXF router. In other words, the nearest ICXF is the default router for all external IPv4 prefixes.
- o Pros: Significantly reduces the size of the IPv6 routing table to virtually one entry for IPv4 reachability.
- o Cons: The closest ICXF router may not have the best route to the final destination in the IPv4 network. The ICXF may forward the packet to another ICXF to reach the IPv4 destination based upon the local IPv4 routing information.

Option 4:

- o This option requires every router in the IPv6 network to learn the IPv4-Embedded IPv6 prefixes advertised by the AFTR and ICXF. These prefixes are only meaningful to the AFTR and ICXF, other IPv6 routers are not interested in them. To address this issue, a new topology [RFC4915] or [RFC5120] can be created to store the IPv4-Embedded IPv6 prefixes.
- o This option requires the ICXF router and AFTR advertise the IPv4-Embedded IPv6 prefixes in the IPv4-Embedded IPv6 topology. This topology contains only the IPv4-Embedded IPv6 prefixes. Regular IPv6 routers will not participate this topology.

- o With this option, each ICXF router advertises its reachable IPv4 prefixes in the form of the IPv4-Embedded IPv6 addresses. These LSAs will appear only in the dedicated MT. AFTR which participates the MT will install the LSAs to its IPv6 routing table. Those didn't participates the MT will simply ignore the LSAs.
- o Pros: Only the AFTR and ICXF install the IPv4-Embedded IPv6 prefixes in the IPv6 routing table.
- o Cons: Addition administration cost to maintain a new topology in ICXF and AFTR.

7. Multicast Considerations

This document describes an IPv4-IPv6 inter-connection extension to DS-Lite [I-D.ietf-softwire-dual-stack-lite], which currently limits the scope to transporting unicast IPv4 traffic over IPv6 network only. Considerations on transporting multicast IPv4 traffic over IPv6 network is out of scope.

8. Fragmentation

Tunneling IPv4 over IPv6 between AFTR and ICXF reduce the effective MTU size by the size of an IPv6 header. Since ICXF tunnel is stateless, the tunnel endpoint can't fragment and re-assumable the oversized IPv4 packet. A service provider may increase the MTU size by 40-bytes on the IPv6 network. If this is not possible, AFTR and ICXF may use IPv6 Path MTU discovery.

ICXF nodes are stateless and not necessary to implement IPv4 fragmentation.

9. Conclusions

This document describes the mechanism to enable AFTR to operate on an IPv6-only network while offering:

- o Global IPv6 <==> IPv6 communications.
- o Global IPv4 <==> IPv4 communications.
- o A remote IPv6 host would reach a host connected to the DS-Lite enabled domain using IPv6.

- o A remote IPv4 host would reach a host connected to the DS-Lite enabled domain using IPv4-in-IPv6.

10. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

11. Security Considerations

Security considerations defined in [I-D.ietf-softwire-dual-stack-lite] should be taken into account. In addition, current interconnection practices for ingress traffic filtering should be enforced in the interconnection points (ICXF).

12. Acknowledgements

The authors would like to thank Eric Burgey for his support and suggestions.

13. References

13.1. Normative References

- [I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-06 (work in progress), August 2010.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

13.2. Informative References

- [I-D.ietf-behave-address-format]
 Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", draft-ietf-behave-address-format-10 (work in progress), August 2010.
- [RFC4277] McPherson, D. and K. Patel, "Experience with the BGP-4 Protocol", RFC 4277, January 2006.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, June 2007.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, February 2008.
- [RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, June 2009.

Appendix A. Changes Since 02

The following changes have been made since the last version:

1. Add a new section to define addressing scheme;
2. Add a new section to list all routing options in the IPv6 network;
3. Various editorial changes.

Authors' Addresses

Mohamed Boucadair (editor)
France Telecom
3, Av Francois Chateaux
Rennes 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

Christian Jacquenet
France Telecom

Email: christian.jacquenet@orange-ftgroup.com

Jean-Luc Grimault
France Telecom
France

Email: jeanluc.grimault@orange-ftgroup.com

Mohammed Kassi-Lahlou
France Telecom

Email: mohamed.kassilahlou@orange-ftgroup.com

Pierre Levis
France Telecom

Email: pierre.levis@orange-ftgroup.com

Dean Cheng (editor)
Huawei Technologies Co., Ltd.

Email: Chengd@huawei.com
URI: <http://www.huawei.com>

Yiu L. Lee (editor)
Comcast

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

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 21, 2011

F. Brockners
Cisco
Y. Lee
Comcast
October 18, 2010

Multicast Considerations for Gateway-Initiated Dual-Stack Lite
draft-brockners-softwire-mcast-gi-ds-lite-00

Abstract

This document discusses multicast deployment aspects for networks which leverage Gateway-Initiated Dual-Stack lite.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 21, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction and Overview	3
2. Abbreviations	4
3. Multicast Deployment Considerations	4
3.1. Architectural Attributes	4
3.2. Overlapping private IPv4 addresses	5
3.3. Considerations for the Gateway and AFTR	6
4. Acknowledgements	6
5. IANA Considerations	6
6. Security Considerations	7
7. References	7
7.1. Normative References	7
7.2. Informative References	7
Authors' Addresses	8

1. Introduction and Overview

This draft discusses the deployment aspects for IPv4-Multicast in networks using Gateway-Initiated Dual-Stack lite (GI-DS-lite) [I-D.ietf-software-gateway-init-ds-lite]. GI-DS-lite is a modified approach to the original Dual-Stack lite (DS-lite) [I-D.ietf-software-dual-stack-lite] applicable to certain tunnel-based access architectures. Figure 1 shows an example. GI-DS-lite extends existing access tunnels beyond the Gateway to an IPv4-IPv4 NAT device (as shown in Figure 2) using softwires with an embedded context identifier, that uniquely identifies the end-system the tunneled packets belong to. The Gateway determines which portion of the traffic requires NAT using local policies and sends/receives this portion to/from this softwire tunnel.

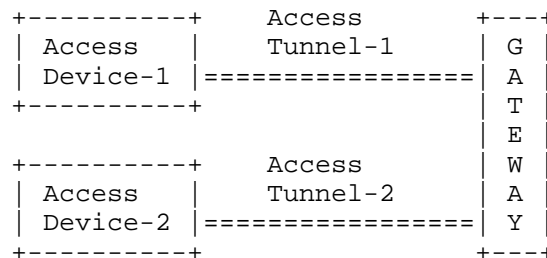


Figure 1: Tunnel based access architecture

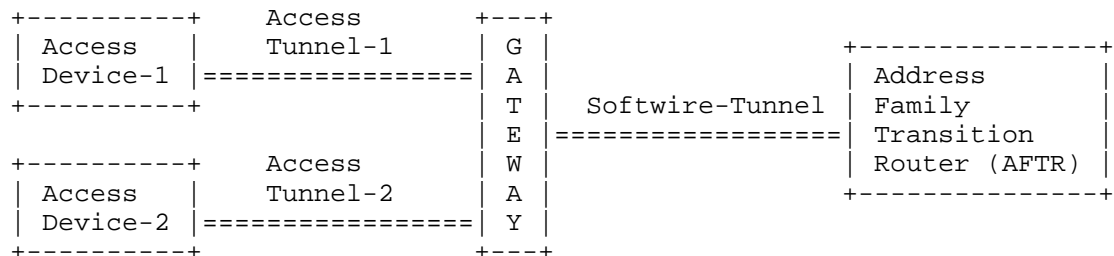


Figure 2: Gateway-initiated dual-stack lite reference architecture

Some applications require multicast to deliver services to the access devices. For example: Live sport event and IP-TV broadcast could use multicast to deliver video streams to the access devices. During IPv4-IPv6 transitioning, the multicast traffic could continue to be transported over IPv4, access devices behind GI-DS-lite require an architecture to subscribe to IPv4-Multicast groups and receive IPv4-

Multicast traffic. Currently, most IPv4-Multicast deployments require the access devices to receive multicast traffic but not to source multicast traffic. This memo considers the scenario where the access device subscribes an IPv4-Multicast group and recommends how the multicast routing could be done. The following cases are out of scope in this memo:

- o IPv4-Multicast sourced by Access Devices.
- o Network Address Translation (NAT) for IPv4-Multicast traffic.

2. Abbreviations

The following abbreviations are used within this document:

AFTR: Address Family Transition Router (also known as "Large Scale NAT (LSN)" or "Dual-Stack lite Tunnel Concentrator", or "Carrier Grade NAT (CGN)"). An AFTR combines IP-in-IPv6 tunnel termination and IPv4-IPv4 NAT.

DS-lite: Dual-stack lite

GI-DS-lite: Gateway-initiated DS-lite

NAT: Network Address Translation

3. Multicast Deployment Considerations

This section details the IPv4-Multicast deployment considerations for GI-DS-lite. Several networks which follow the architecture shown in Figure 1 above deploy IPv4-Multicast. If GI-DS-lite is introduced, the Gateway continues to perform the role of the first hop IPv4-Multicast router and the overall multicast distribution architecture is left unchanged. Deployment dependent, the introduction of GI-DS-lite could go hand in hand with the Gateway no longer having native IPv4-Multicast connectivity. If the Gateway does not have native IPv4-Multicast connectivity it should create a tunnel (e.g. IP-in-IPv6 or IP-over-GRE6) to an IPv4-Multicast router (e.g. the closest). The Gateway peers with that IPv4-Multicast router via the tunnel to join the IPv4-Multicast routing domain.

3.1. Architectural Attributes

Deployment details for IP-Multicast are defined for several architectures which leverage tunnel-based access, such as [TR101] for DSL-Broadband, 3GPP TS 23.246 for mobile Multimedia Broadcast Service

(MBMS) [TS23246], or [I-D.ietf-multimob-pmipv6-base-solution] for multicast in Proxy Mobile-IP deployments). Multicast in mobile or broadband deployments with tunnel based access architectures share a set of common architectural attributes:

- o Subscribers are able to receive IP multicast, but are not assumed to send IP multicast (inline with the scope of this document).
- o The Gateway is an IP-Multicast router, which is attached to the IP-Multicast distribution network of the service provider.
- o Architectures often include devices which perform IGMP/MLD snooping and proxy reporting between the access device and the Gateway. Proxy Mobile IPv6 deployments [I-D.ietf-multimob-pmipv6-base-solution] are an example: Mobile devices (i.e. the Access Devices) are connected via a Mobile Access Gateway (MAG) implementing an IGMP proxy function to the Local Mobility Anchor (LMA) which performs the role of the Gateway.
- o In several broadband multicast deployments IP-Multicast traffic is not forwarded over the access tunnels used for unicast traffic, but uses an alternate vehicle, which allows for traffic replication between access devices and Gateway. DSL-broadband networks with Ethernet aggregation are an example: While unicast traffic is forwarded between the access devices and the Broadband Network Gateway (BNG) over dedicated point to point VLANs, a separate VLAN is used to forward multicast traffic. This allows taking advantage of the multicast replication capabilities of Ethernet within the aggregation network.

3.2. Overlapping private IPv4 addresses

GI-DS-lite supports deployments with (potentially overlapping) IPv4 addresses assigned to the access devices. This could present challenges from a theoretical point of view for the following scenarios:

1. The network deploys Source Specific Multicast (SSM) and IP-multicast is sourced from an access device: Per the note above, this scenario is out of the scope of this document.
2. The network deploys IGMPv3 and leverages explicit tracking (see appendix 2 of [RFC3376], or appendix 2 of [RFC3810]) only based on the source IP address of the IGMP messages: Explicit tracking is in use by several networks today, though one often does not rely (only) on the source IP-address to identify different hosts. Several multicast networks deploy devices performing IGMP

snooping with proxy reporting between the multicast host and the first hop IP-Multicast router. In those deployments, the source IP address of the IGMP join messages does no longer represent the multicast host. The Broadband Forum, for example, requires the source IP address of IGMP packets sent by the proxy reporting function to be 0.0.0.0 [TR101]. If explicit tracking is still desired in those environments, identifiers other than the source IP address need to be considered. Depending on the deployment and architecture, those could for example be the interface (as recommended for proxy Mobile-IP multicast deployments [I-D.ietf-multimob-pmipv6-base-solution]) or the MAC-address (see [TR101]), an access tunnel identifier etc.).

3.3. Considerations for the Gateway and AFTR

The Gateway's role with regards to IPv4-Multicast traffic forwarding and routing does not change if GI-DS-lite is deployed within the network and multicast traffic bypasses the AFTR. For deployments which require explicit tracking and the use of overlapping IPv4 address ranges at a Gateway, this Gateway needs to support explicit tracking based on identifiers other than the source IP-address of IGMP messages. The Gateway functions as IPv4-Multicast first hop router for the access devices. The Gateway is a multicast replication point for multicast flows towards receivers on or attached to the access devices. IPv4-Multicast traffic will be forwarded according to the group/source-specific forwarding states. If there are multiple receivers within the scope of the Gateway, its still a single flow which the Gateway receives. IPv4-Multicast forwarding at the Gateway is also not impacted in case IGMP-proxies exist between the access devices and the Gateway. This can be the case in broadband architectures (see [TR101]) as well as in mobile architectures (e.g., with PMIP, the MAG acts as MLD proxy, see [I-D.ietf-multimob-pmipv6-base-solution]).

4. Acknowledgements

The authors would like to acknowledge their discussions on this topic with Wojciech Dec and Sri Gundavelli.

5. IANA Considerations

This document includes no request to IANA.

6. Security Considerations

This draft does not introduce additional messages or novel protocol operations. Consequently, no new threats are introduced by this document in addition to those identified as security concerns for IP-Multicast deployments.

7. References

7.1. Normative References

- [I-D.ietf-multimob-pmipv6-base-solution]
Schmidt, T., Waehlich, M., and S. Krishnan, "Base Deployment for Multicast Listener Support in PMIPv6 Domains", draft-ietf-multimob-pmipv6-base-solution-05 (work in progress), July 2010.
- [I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-06 (work in progress), August 2010.
- [I-D.ietf-softwire-gateway-init-ds-lite]
Brockners, F., Gundavelli, S., Speicher, S., and D. Ward, "Gateway Initiated Dual-Stack Lite Deployment", draft-ietf-softwire-gateway-init-ds-lite-00 (work in progress), May 2010.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.

7.2. Informative References

- [TR101] Broadband Forum, "TR-101: Migration to Ethernet-Based DSL Aggregation", April 2006.
- [TS23246] 3GPP, "3GPP TS 23.246: Multimedia Broadcast/Multicast

Service (MBMS), Architecture and functional description,
Release 9", June 2010.

Authors' Addresses

Frank Brockners
Cisco
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

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

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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2012

Y. Cui
J. Wu
P. Wu
Tsinghua University
C. Metz
Cisco Systems, Inc.
O. Vautrin
Juniper Networks
Y. Lee
Comcast
July 8, 2011

Public IPv4 over Access IPv6 Network
draft-cui-softwire-host-4over6-06

Abstract

This draft proposes a mechanism for bidirectional IPv4 communication between IPv4 Internet and end hosts or IPv4 networks sited in IPv6 access network. This mechanism follows the softwire hub and spoke model and uses IPv4-over-IPv6 tunnel as basic method to traverse IPv6 network. By allocating public IPv4 addresses to end hosts/networks in IPv6, it can achieve IPv4 end-to-end bidirectional communication between these hosts/networks and IPv4 Internet. This mechanism is an IPv4 access method for hosts and IPv4 networks sited in IPv6.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 9, 2012.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Requirements language	4
3. Terminology	5
4. Deployment scenario	6
4.1. Scenario and requirements	6
4.2. Use cases	7
5. Public 4over6 Mechanism	9
5.1. Address allocation and mapping maintenance	9
5.2. 4over6 initiator behavior	9
5.2.1. Host initiator	10
5.2.2. CPE initiator	10
5.3. 4over6 concentrator behavior	11
6. Technical advantages	12
7. Acknowledgement	13
8. References	14
8.1. Normative References	14
8.2. Informative References	14
Authors' Addresses	16

1. Introduction

Global IPv4 addresses are running out fast. Meanwhile, the demand for IP address is still growing and may even burst in potential circumstances like "Internet of Things". To satisfy the end users, operators have to push IPv6 to the front, by building IPv6 networks and providing IPv6 services.

When IPv6-only networks are widely deployed, users of those networks will probably still need IPv4 connectivity. This is because part of Internet will stay IPv4-only for a long time, and network users in IPv6-only networks will communicate with network users sited in the IPv4-only part of Internet. This demand could eventually decrease with the general IPv6 adoption.

Network operators should provide IPv4 services to IPv6 users to satisfy their demand, usually through tunnels. This type of IPv4 services differ in provisioned IPv4 addresses. If the users can't get public IPv4 addresses (e.g., new network users join an ISP which don't have enough unused IPv4 addresses), they have to use private IPv4 addresses on the client side, and IPv4-private-to-public translation is required on the carrier side, as is described in Dual-stack Lite[I-D.ietf-softwire-dual-stack-lite]. Otherwise the users can get public IPv4 addresses, and use them for IPv4 communication. In this case, translation on the carrier side won't be necessary. The network users and operators can avoid all the issues raised by translation, such as ALG, NAT traversal, state maintenance, etc. Note that this "public IPv4" situation is actually quite common. There're approximatively 2^{32} network users who are using or can potentially get public IPv4 addresses. Most of them will switch to IPv6 sooner or later, and will require IPv4 services for a significant period after the switching. This draft focuses on this situation, i.e., to provide IPv4 access for users in IPv6 networks, where public IPv4 addresses are still available for allocation.

2. Requirements language

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

3. Terminology

Public 4over6: Public 4over6 is the mechanism proposed by this draft. Generally, Public 4over6 supports bidirectional communication between IPv4 Internet and IPv4 hosts or local networks in IPv6 access network, by leveraging IPv4-in-IPv6 tunnel and public IPv4 address allocation.

4over6 initiator: in Public 4over6 mechanism, 4over6 initiator is the IPv4-in-IPv6 tunnel initiator located on the user side of IPv6 network. The 4over6 initiator can be either a dual-stack capable host or a dual-stack CPE device. In the former case, the host has both IPv4 and IPv6 stack but is provisioned with IPv6 access only. In the latter case, the CPE has both IPv6 interface for access to ISP network and IPv4 interface for local network connection; hosts in the local network can be IPv4-only.

4over6 concentrator: in Public 4over6 mechanism, 4over6 concentrator is the IPv4-in-IPv6 tunnel concentrator located in IPv6 ISP network. It's a dual-stack router which connects to both the IPv6 network and IPv4 Internet.

4. Deployment scenario

4.1. Scenario and requirements

The general scenario of Public 4over6 is shown in Figure 1. Users in an IPv6 network take IPv6 as their native service. Some users are end hosts which face the ISP network directly, while others are local networks behind CPEs, such as a home LAN, an enterprise network, etc. The ISP network is IPv6-only rather than dual-stack, which means that ISP can't provide native IPv4 access to its users; however, it's acceptable that one or more routers on the carrier side become dual-stack and get connected to IPv4 Internet. So if network users want to connect to IPv4, these dual-stack routers will be their "entrances".

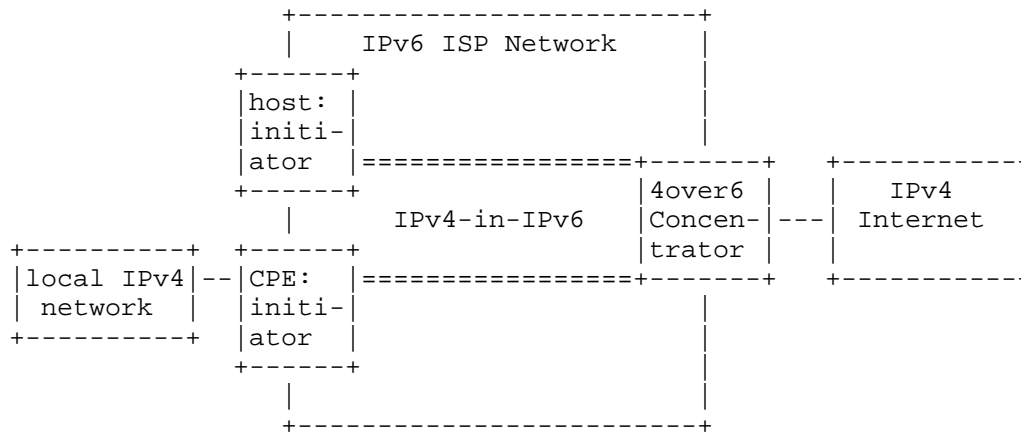


Figure 1 Public 4over6 scenario

Before getting into any technical details, the communication requirements should be stated. The first one is that, 4over6 users require IPv4-to-IPv4 communication with the IPv4 Internet. An IPv4 access service is needed rather than an IPv6-to-IPv4 translation service. (IPv6-to-IPv4 communication is out of the scope of this draft.)

Second, 4over6 users require public IPv4 addresses rather than private addresses. Public IPv4 address means there's no IPv4 CGN along the path, so the acquired IPv4 service is better. In particular, some hosts may be application servers, public address works better for reasons like straightforward access, direct DNS registration, no stateful mapping maintenance on CGN, etc. For the

direct-connected host case, each host should get one public IPv4 address. For the local IPv4 network case, the CPE can get a public IPv4 address and runs an IPv4 NAT for the local network. Here a local NAT is still much better than the situation that involves a CGN, since this NAT is in local network and can be configured and managed by the users.

Third, translation is not preferred in this scenario. If this IPv4-to-IPv4 communication is achieved by IPv4-IPv6 translation, it'll need double translation along the path, one from IPv4 to IPv6 and the other from IPv6 back to IPv4. This would be quite complicated, especially in addressing. Contrarily a tunnel can achieve the IPv4-over-IPv6 traversing easily. That's the reason this draft follows the hub and spoke software model.

Moreover, the ISP probably would like to keep their IPv4 and IPv6 addressing and routing separated when provisioning IPv4 over IPv6. Then the ISP can manage the native IPv6 network more easily and independently, and also provision IPv4 in a flexible, on-demand way. The cost is that the concentrator needs to maintain per-user address mapping state, which would be described in detail.

4.2. Use cases

Public 4over6 can be applicable in several practical cases. The first one is that ISPs which still own enough IPv4 addresses switch to IPv6. The ISPs can deploy public 4over6 to preserve IPv4 service for the customers. This case is actually quite common. The majority of the wired end users today get Internet access with public IPv4 address. When their ISPs switch to IPv6, these users can still use the same amount of IPv4 addresses for IPv4 access. Public 4over6 can leveraging these addresses and offer tunneled IPv4 access.

The second case is ISPs which don't have enough IPv4 addresses any more switch to IPv6. For these ISPs, dual-stack lite is so far the most mature solution to provision IPv4 over IPv6. In dual-stack lite, end users use private IPv4 addresses, experience a 4CGN and hence some service degradation. As long as the end users use public IPv4 addresses, all CGN issues can be avoided and the IPv4 service can be full bi-directional. In other words, Public 4over6 can be deployed along with DS-lite, to provide a value-added service. Common users adopt DS-lite to communicate with IPv4 while high-end users adopt Public 4over6. The two mechanisms can actually be coupled easily.

There is also a special situation in the second case that the end users are IPv4 application servers. In this situation, public address brings significant convenience. The DNS registration can be

direct using dedicated address; the access of application clients can be straightforward with no translation; there's no need to reserve and maintain address mapping on the CGN, and no well-known port collision will come up. So it's better to have servers adopt Public 4over6 for IPv4 access when they're located in IPv6 network.

Following the principle of Public 4over6, it's also possible to achieve address multiplexing and save IPv4 addresses. There're already efforts on this subject, see [I-D.cui-software-b4-translated-ds-lite] and [I-D.sun-v6ops-laft6]. The basic idea is that instead of allocating a full IPv4 address to every end user, the ISP can allocate an IPv4 address with restricted port range to every end user.

Besides, the draft would like to be explicit about the scope of direct-connected host case and CPE case. The host case is clear: the host is directly connected to IPv6 network, but the protocol stack on the host support IPv4 too. As to the CPE case, this draft would like to only focus on the case that the local network behind the CPE is private IPv4. If the users want to run public IPv4 into the local network, then they can either run dual-stack in the local network and turn into host case(likely home LAN situation), or they can acquire address blocks from the ISP and build configured tunnel or software mesh[RFC5565] with the ISP network(likely enterprise network situation). TC can be implemented to be compatible with the latter case too, though.

5. Public 4over6 Mechanism

5.1. Address allocation and mapping maintenance

Public 4over6 can be generally considered as IPv4-over-IPv6 hub and spoke tunnel using public IPv4 address. Each 4over6 initiator will use public IPv4 address for IPv4-over-IPv6 communication. As is described above, in the host initiator case, every host will get one IPv4 address; in the CPE case, every CPE will get one IPv4 address, which will be shared by hosts behind the CPE. The key problem here is IPv4 address allocation over IPv6 network, from ISP device(s) to separated 4over6 initiators.

There're two possibilities here. One is DHCPv4 over IPv6, and the other is static configuration. DHCPv4 over IPv6 is achieved by performing DHCPv4 on IPv4-in-IPv6 tunnel between ISP device and 4over6 initiators. There do exist the DHCP encapsulation issue on server side, see details and solutions in [I-D.cui-software-dhcp-over-tunnel]. As to static configuration, 4over6 users and the ISP operators should negotiate beforehand to authorize the IPv4 address. Application servers usually falls into this case. Public 4over6 supports both address allocation manners. Actually, it is transparent to address allocation methods.

Along with IPv4 address allocation, Public 4over6 should maintain the IPv4-IPv6 address mappings on the concentrator. In this type of address mapping, the IPv4 address is the public IPv4 address allocated to a 4over6 initiator, and the IPv6 addresses is the initiator's IPv6 address. This mapping is used to provide correct encapsulation destination address for the concentrator.

The initiator sends "pinhole" packets to the concentrator periodically, to install and renew the address mapping. A pinhole packet is an IPv4-in-IPv6 packet, which uses the concentrator's IPv6 address as destination IPv6 address, the initiator's IPv6 address as source IPv6 address, and the initiator's IPv4 address as source IPv4 address. When the concentrator receives such a packet, it'll resolve the IPv4 and IPv6 address information from the packet and trigger the mapping. Since any IPv4-in-IPv6 data packet from the initiator contains these exact informations, it can also serve as pinhole packet. Then dedicated pinhole packets are sent out when there's no data packets. Another possible way to maintain the address mapping is to run PCP[I-D.ietf-pcp-base] while extending the protocol to support applying for a full address. The following sections describe the mechanism with the pinhole method.

5.2. 4over6 initiator behavior

4over6 initiator has an IPv6 interface connected to the IPv6 ISP network, and a tunnel interface to support IPv4-in-IPv6 encapsulation. In CPE case, it has at least one IPv4 interface connected to IPv4 local network.

4over6 initiator should learn the 4over6 concentrator's IPv6 address beforehand. For example, if the initiator gets its IPv6 address by DHCPv6, it can get the 4over6 concentrator's IPv6 address through a DHCPv6 option[I-D.ietf-softwire-ds-lite-tunnel-option].

5.2.1. Host initiator

When the initiator is a direct-connected host, it assigns the allocated public IPv4 address to its tunnel interface. The host uses this address for IPv4 communication. If this address is allocated through DHCP, the host should support DHCPv4 over tunnel. After the allocation, the host periodically sends pinhole packet to the concentrator to install the address mapping and keep it alive.

For IPv4 data traffic, the host performs the IPv4-in-IPv6 encapsulation and decapsulation on the tunnel interface. When sending out an IPv4 packet, it performs the encapsulation, using the IPv6 address of the 4over6 concentrator as the IPv6 destination address, and its own IPv6 address as the IPv6 source address. The encapsulated packet will be forwarded to the IPv6 network. The decapsulation on 4over6 initiator is simple. When receiving an IPv4-in-IPv6 packet, the initiator just drops the IPv6 header, and hands it to upper layer.

5.2.2. CPE initiator

The CPE case is quite similar to the host initiator case. The CPE assign the allocated IPv4 address to its tunnel interface. The local IPv4 network won't take part in the public IPv4 allocation; instead, end hosts will use private IPv4 addresses, possibly allocated by the CPE. After the allocation, the CPE periodically sends pinhole packet to the concentrator to install the address mapping and keep it alive.

On data plan, the CPE can be viewed as a regular IPv4 NAT(using tunnel interface as the NAT outside interface) cascaded with a tunnel initiator. For IPv4 data packets received from the local network, the CPE translates these packets, using the tunnel interface address as the source address, and then encapsulates the translated packet into IPv6, using the concentrator's IPv6 address as the destination address, the CPE's IPv6 address as source address. For IPv6 data packet received from the IPv6 network, the CPE performs decapsulation and IPv4 public-to-private translation. As to the CPE itself, it uses the public, tunnel interface address to communicate with the

IPv4 Internet, and the private, IPv4 interface address to communicate with the local network.

5.3. 4over6 concentrator behavior

4over6 concentrator represents the IPv4-IPv6 border router working as the remote tunnel endpoint for 4over6 initiators, with its IPv6 interface connected to the IPv6 network, IPv4 interface connected to the IPv4 Internet, and a tunnel interface supporting IPv4-in-IPv6 encapsulation and decapsulation. There's no CGN on the 4over6 concentrator, it won't perform any translation function; instead, 4over6 concentrator maintains an IPv4-IPv6 address mapping table for IPv4 data encapsulation.

4over6 concentrator maintains the address mapping according to the initiators' demand. When receiving a pinhole packet from an initiator, the concentrator reads the IPv4 and IPv6 source addresses from the packet, install the mapping entry into the mapping table or renew it if it already exists. When the lifetime of a mapping entry expires, the concentrator deletes it from the table. So the initiator should send pinhole packet with an interval shorter than the lifetime of the mapping entry. The mapping entry is used to provide correct encapsulation destination address for concentrator encapsulation. As long as the entry exists in the table, the concentrator can encapsulate inbound IPv4 packets destined to the initiator, with the initiator's IPv6 address as IPv6 destination.

On the IPv6 side, 4over6 concentrator decapsulates IPv4-in-IPv6 packets coming from 4over6 initiators. It removes the IPv6 header of every IPv4-in-IPv6 packet and forwards it to the IPv4 Internet. On the IPv4 side, the concentrator encapsulates the IPv4 packets destined to 4over6 initiators. When performing the IPv4-in-IPv6 encapsulation, the concentrator uses its own IPv6 address as the IPv6 source address, uses the IPv4 destination address in the packet to look up IPv6 destination address in the address mapping table. After the encapsulation, the concentrator sends the IPv6 packet on its IPv6 interface to reach an initiator.

The 4over6 concentrator, or its upstream router should advertise the IPv4 prefix which contains the IPv4 addresses of 4over6 users to the IPv4 side, in order to make these initiators reachable on IPv4 Internet.

Since the concentrator has to maintain the IPv4-IPv6 address mapping table, the concentrator is stateful in IP level. Note that this table will be much smaller than a CGN table, as there is no port information involved.

6. Technical advantages

Public 4over6 provides a method for users in IPv6 network to communicate with IPv4. In many scenarios, this can be viewed as an alternative to IPv6-IPv4 translation mechanisms which have well-known limitations described in [RFC4966] .

Since a 4over6 initiator uses a public IPv4 address, Public 4over6 supports full bidirectional communication between IPv4 Internet and hosts/IPv4 networks in IPv6 access network. In particular, it supports the servers in IPv6 network to provide IPv4 application service transparently.

Public 4over6 provides IPv4 access over IPv6 network while keeps IPv4-IPv6 addressing and routing separated. Therefore the ISP can manage the native IPv6 network independently without the influence of IPv4-over-IPv6 requirements, and also provision IPv4 in a flexible, on-demand way.

Public 4over6 supports dynamic reuse of a single IPv4 address between multiple subscribers based on their dynamic requirement of communicating with IPv4 Internet. A subscriber will request a public IPv4 address for a period of time only when it need to communicate with IPv4 Internet. Besides, in the CPE case, one public IPv4 address will be shared by the local network. So Public 4over6 can improve the reuse rate of IPv4 addresses.

Public 4over6 is suited for network users/ISPs which can still get/provide public IPv4 addresses. Dual-stack lite is suited for network users/ISPs which can no longer get/provide public IPv4 addresses. By combining Public 4over6 and Dual-stack lite, the IPv4-over-IPv6 Hub and spoke problem can be well solved.

7. Acknowledgement

The authors would like to thank Alain Durand and Dan Wing for their valuable comments on this draft.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4925] Li, X., Dawkins, S., Ward, D., and A. Durand, "Softwire Problem Statement", RFC 4925, July 2007.
- [RFC4966] Aoun, C. and E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", RFC 4966, July 2007.
- [RFC5549] Le Faucheur, F. and E. Rosen, "Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop", RFC 5549, May 2009.
- [RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, June 2009.

8.2. Informative References

- [I-D.cui-softwire-b4-translated-ds-lite]
Cui, Y., Wu, J., and D. Wu, "B4 translated DS-lite enable AFTR to serve more B4s",
draft-cui-softwire-b4-translated-ds-lite-00 (work in progress), October 2010.
- [I-D.cui-softwire-dhcp-over-tunnel]
Cui, Y., Wu, P., and J. Wu, "DHCPv4 Behavior over IP-IP tunnel", draft-cui-softwire-dhcp-over-tunnel-00 (work in progress), June 2011.
- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)",
draft-ietf-pcp-base-13 (work in progress), July 2011.
- [I-D.ietf-softwire-ds-lite-tunnel-option]
Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual- Stack Lite",
draft-ietf-softwire-ds-lite-tunnel-option-10 (work in progress), March 2011.
- [I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4

Exhaustion", draft-ietf-softwire-dual-stack-lite-11 (work in progress), May 2011.

[I-D.sun-v6ops-laft6]

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

Authors' Addresses

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

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

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

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

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

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

Chris Metz
Cisco Systems, Inc.
3700 Cisco Way
San Jose, CA 95134
USA

Email: chmetz@cisco.com

Olivier Vautrin
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, CA 94089
USA

Email: Olivier@juniper.net

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

Email: yiul_lee@cable.comcast.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 28, 2011

Y. Cui
M. Xu
P. Wu
S. Wang
J. Wu
X. Li
Tsinghua University
C. Metz
Cisco Systems, Inc.
October 25, 2010

Translation Spot Negotiation in IPv4/IPv6-Coexist Mesh
draft-cui-softwire-pet-03

Abstract

IPv4 and IPv6 are expected to coexist for a long period. Currently, there are many IPv4/IPv6 transition/coexistence techniques, roughly divided into the categories of tunneling and translation. Tunneling and translation have respective application scopes, and translation has some technical limitations, including scalability issue, application layer translation, operation complexity, etc. To improve the availability of translation, this draft proposes the method of selecting appropriate translation spot to execute translation. When the translation spot is not on IPv4-IPv6 border, tunnel is used to achieve the traversing between translation spot and IP border. This method applies well in mesh scenario where both IPv4 and IPv6 client network exists, and BGP can be extended to achieve a translation spot signaling.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 28, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

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

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
2. Translation Spot Selection	6
3. Translation Spot Selection in IPv4/IPv6-coexist Mesh	8
3.1. Scenario description	8
3.2. Translation between IPvX and IPvY networks	9
3.3. Translation between IPvX network and IPvY Internet	9
3.4. Translation between IPvY network and IPvX Internet	9
4. Translation Spot Signaling	10
4.1. Signaling content	10
4.2. Extensions in MP-BGP	10
5. Further discussion	13
5.1. Achievement of translation spot selection	13
5.2. Cooperate with softwire	13
5.3. Using NAT64 or IVI as translation mechanism	13
6. IANA considerations	14
7. Acknowledgements	15
8. References	16
8.1. Normative References	16
8.2. Informative References	17
Authors' Addresses	18

1. Introduction

Recently more and more IPv6 networks have been deployed, especially IPv6 backbone networks. However the existing IPv4 networks still carry the major network traffic and hold the major network services and applications. It has been widely believed that IPv4 and IPv6 networks will coexist for a long term. This leads to the demand for IPv4-IPv6 coexistence technology.

Till now there are two types of IPv4-IPv6 coexistence techniques: tunneling and translation. Tunneling can achieve IPv4-over-IPv6/IPv6-over-IPv4 traversing, by means of encapsulation and decapsulation. Examples of tunneling methods include IP-in-IP tunnel [RFC2893][RFC4213], GRE tunnel [RFC1702], 6to4 tunnel [RFC3056], 6over4 tunnel [RFC2529], softwire mesh technique [RFC5565], etc. Tunneling is transparent and light-weighted. It can be implemented fully by hardware.

On the other hand, translation is used to achieve IPv4-IPv6 inter-communication, by means of converting the semantic between IPv4 and IPv6. Examples of translation methods include SIIT [RFC2765], NAT-PT [RFC2766], BIS [RFC2767], BIA [RFC3338], IVI [I-D.xli-behave-ivil], NAT64 [I-D.ietf-behave-v6v4-xlate-stateful] and so on. Translation can achieve IPv4-IPv6 interworking which tunneling cannot do, but it has several technical limitations:

- o Scalability. In stateful translation, the dynamic mapping of (address, port) tuple should be maintained on the translation device. The total number of mapping entries is up to the order of flow number. As to stateless translation, it has to consume IPv4 addresses to satisfy IPv6 hosts. This is also not scalable since IPv6 address space is much larger than IPv4 address.
- o Application layer translation. Since translation will modify the address of an IP packet, or we say an end host, an application protocol that contains IP addresses in its payload won't work if we don't convert the addresses. However, due to the variety of applications protocols, it's unrealistic for the translation device to support all of them.
- o Operation complexity. To accomplish correct translation, the following operations are required: address or (address, port) tuple conversion, IP and ICMP fields translation, TCP/UDP checksum re-computing, application layer detection and translation, fragmentation when necessary. It's rather complex for a per-packet process and probably unacceptable when the volume is high.

- o Lack of efficient NAT46 translation mechanism. No efficient IPv4 to IPv6 communication mechanism has been proposed since NAT-PT. A fundamental difficulty here is that IPv6 address space is much larger than IPv4 so the translation mechanism has to make DNS or other addressing method stateful. Obviously this is not scalable.

Though facing all these issues, translation is irreplaceable in its application scope, so it's necessary to find a way to improve its availability. To solve this problem, this draft proposes the method of finding the appropriate translation spot to execute translation. The method adopts tunnel when necessary, to achieve traversing between translation spot and IP border. As an attempt, this draft applies the method in IPv4/IPv6-coexist mesh scenario, and extends BGP to achieve translation spot signaling in the scenario.

2. Translation Spot Selection

The issues of translation listed in section 1 are inherent disadvantages due to the principle of translation. Hence it's difficult to solve these problems by improving the mechanism. However, by choosing the appropriate location to perform translation, these problems can be solved or lightened, and translation can be more available. This draft calls the location to perform translation as "translation spot".

The basic idea of translation spot selection is to choose the place where the scalability and complexity is not a concern, i.e., the place where the translator is capable for its own translation traffic. Following this thought, a straightforward principle is to push translation down to edge networks. Since the volume of translation traffic in edge networks is relatively low, it's possible to achieve a real-time per-flow mapping and per-packet modification there. On the contrary, traffic in backbone is aggregated and hence much higher in volume. So routers in backbone would rather only support routing and forwarding than take charge of high-speed translation. However, when the total translation volume is low, it's easier to perform a unified translation in backbone than to distribute the job to many edge networks.

To achieve flexible translation spot selection, there's still a difficulty in packet forwarding: in a given topology, the IPv4-IPv6 border spot is fixed; If the translation spot isn't identical to the IP border spot, the packets can't be forwarded between the two spot due to IP diversity. See the example in Figure 1. The IP border is on spot 2 between IPvY backbone and IPvX Internet while the translation spot can be spot 1 or spot 2. If spot 1 is chosen, then packets from IPvY edge network are translated into IPvX on spot 1; they have to traverse to IPvY backbone to reach IPvX Internet. , and packets from IPvX Internet have to traverse the IPvY backbone to reach spot 1 for translation. Similar thing happens when spot 2 is chosen in Figure 2.

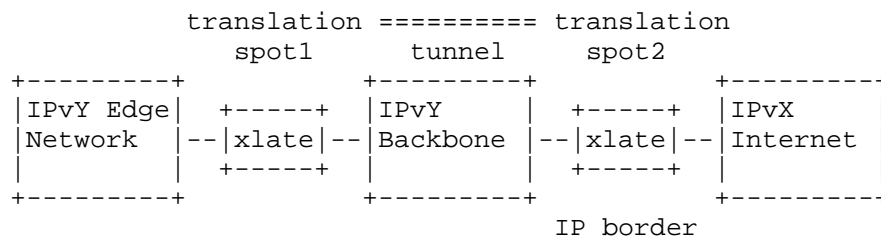


Figure 1 Translation Spot selection

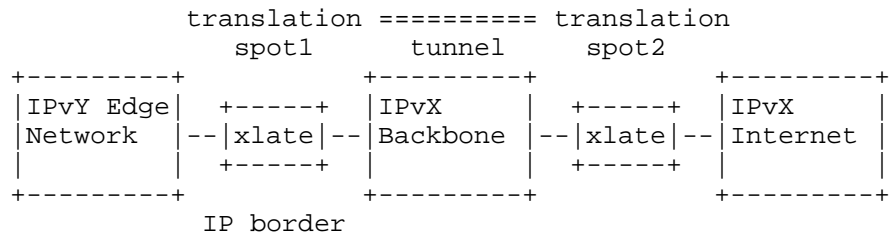


Figure 2 Translation Spot selection

This is actually a traversing problem and the typical solution is tunneling. By building a tunnel to connect IP border and the translation spot, the forwarding path can be achieved. In the example of Figure 1, an IPvX-over-IPvY tunnel between spot 1 and spot 2 can be used to forward translated-to-IPvX packets from spot 1 to spot 2, and to-be-translated IPvX packets from spot 2 to spot 1. In Figure 2, an IPvY-over-IPvX tunnel between spot 1 and spot 2 can be used to forward to-be-translated IPvY packets from spot 1 to spot 2, and translated-to-IPvY packet from spot 2 to spot 1. Although the flexible translation spot selection may require an extra tunnel, its cost is much lower than translation, and hence acceptable.

3. Translation Spot Selection in IPv4/IPv6-coexist Mesh

3.1. Scenario description

Translation spot selection can be used in many scenarios. As a demonstration this draft applies it to the mesh scenario described in Figure 3. In this scenario, an IPvX-only backbone is connected to both IPvX networks and IPvY networks. The backbone may also have entrance to IPvX and IPvY Internet. Besides native traffic and IPvY-over-IPvX software traffic described in [RFC4925], there're also traffics between IPvX and IPvY networks, between IPvX network and IPvY Internet, and between IPvY network and IPvX Internet. All these three types of traffics require translation, which should be performed on AFBRs (Address Family Border Router) or BRs (Border Router) on the border of the backbone.

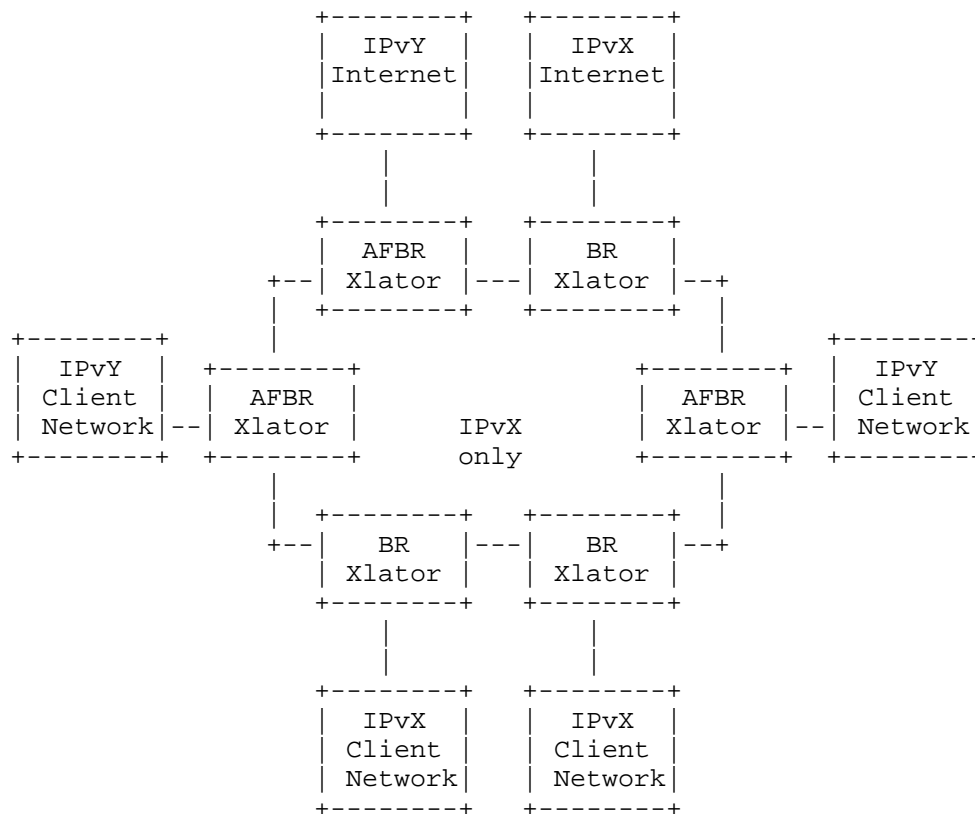


Figure 3 Translation Spot Selection in IPv4/IPv6-coexist Mesh

3.2. Translation between IPvX and IPvY networks

The communication between an IPvX network and an IPvY network follows the path "IPvX network - BR - IPvX backbone - AFBR - IPvY network". The translation can be performed either on the BR between IPvX network and backbone, or on the AFBR between IPvX backbone and IPvY network.

If the BR is chosen to be translation spot, a tunnel should be established for packet forwarding between the BR and the AFBR. Naturally it could be a softwire tunnel since it's a mesh scenario. Besides, to perform correct translation, BR needs the translation context delivered from the AFBR. This will be discussed in the next section.

3.3. Translation between IPvX network and IPvY Internet

The communication between an IPvX network and IPvY Internet follows the path "IPvX network - BR - IPvX backbone - AFBR - IPvY Internet". The translation spot can be either the BR between IPvX network and backbone, or the AFBR between IPvX backbone and IPvY Internet. BR can be chosen to avoid scalability and operation complexity issues, and AFBR can be chosen for unified translation purpose.

If the BR is chosen to be translation spot, a softwire tunnel should be established between the BR and the AFBR. Also BR needs the translation context delivered from the AFBR.

3.4. Translation between IPvY network and IPvX Internet

The communication between an IPvY network and IPvX Internet follows the path "IPvY network - AFBR - IPvX backbone - BR - IPvX Internet". The translation spot can be either the AFBR between IPvY network and IPvX backbone, or the BR between IPvX backbone and IPvX Internet. Usually the AFBR is preferred in this case, since it's the IP border and traffic is not so aggregated as in BR. However, BR can be chosen for unified translation purpose.

If the BR is chosen to be translation spot, a softwire tunnel should be established between the BR and the AFBR. Also BR needs the translation context delivered from the AFBR.

In all three types of translation-involved communication, translation spot selection is feasible. Yet an auto negotiation method is required to make the translation spot selection and translation context advertisement process more practical in the mesh scenario. This will be discussed in the next section.

4. Translation Spot Signaling

In the IPv4/IPv6-coexist mesh, the total number of client networks, and hence the total number of AFBRs and BRs could be quite high, so an auto negotiation method is required to select the translation spot for all translation-involved communications, rather than manual configuration on every AFBR and BR. This negotiation method is called translation spot signaling.

4.1. Signaling content

It's clear that translation should be performed on an appropriate translator, or as in this scenario, an AFBR or BR device. Here the concept of Translation Preference (TP) is defined to represent the appropriateness of a device to perform translation. TP is a quantified value set by the administrator of the corresponding AFBR or BR device. By exchanging and comparing TP values, two translators can decide which one to be the translation spot.

The TP value should be decided by the administrator. The general criterion here is, the translator whose performance is better, whose traffic volume is lower, and the size of network behind which is smaller (thus the translation traffic is less aggregated), is preferred to do translation and should have a high value. TP can also be configured based on administrator's policy, such as unified translation.

Tps for stateless and stateful translation are separated because they have different foundations (stateless translation requires IPv6 host to possess IPv4 address). In a mixed scenario, some translators can't perform stateless translation like others because IPv6 hosts in its network don't own IPv4 addresses.

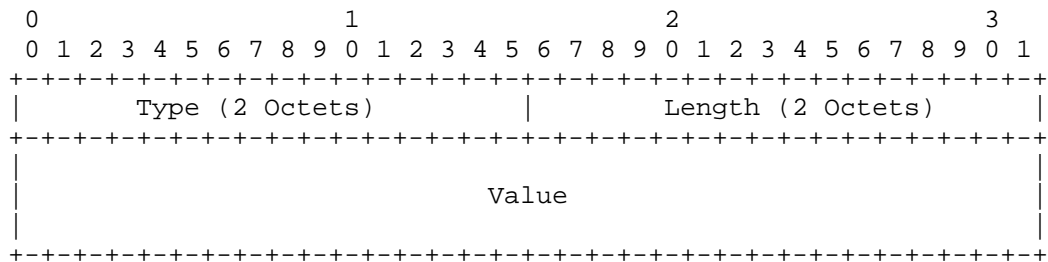
Besides TP, translation context should also be advertised through signaling. The translation context is the necessary knowledge to perform a translation. For stateless translation it's the mapping prefix, and for stateful translation it's the address pool used for address mapping. For example, in the type of "IPv6 network - BR - IPv6 Backbone - AFBR - IPv4 Internet" communication, if stateless translation is adopted, then AFBR should tell BR the prefix for IPv4-IPv6 address mapping when BR performs the translation; if stateful translation is adopted, then AFBR should tell BR the IPv4 addresses BR can use for address mapping when BR performs the translation.

4.2. Extensions in MP-BGP

MP-BGP is adopted to carry the translation spot signaling process since it fits the mesh scenario and is already used in software

mesh[RFC5565].

We define a new a new BGP Attribute, "Translation Information Attribute" to carry the TP and translation context information. It's an optional transitive attribute, and the attribute type code is TBD by IANA. The value field of this attribute is composed of a set of Type-Length-Value (TLV) encodings. The TLV is structured as follows. The Length field stands for the total number of octets in the Value field.



We define 4 TLVs here: Stateless_TP TLV, Stateful_TP TLV, IPv6_Prefix TLV and IPv4_pool TLV. More TLVs may be defined in the future when necessary.

- o Stateless_TP TLV has the type field assigned to 1 and length field assigned to 2. The value field is filled with the 16bit TP value for stateless translation. High the TP value means high preference to perform translation.
- o Stateful_TP TLV has the type field 2 and length field 2. The value field is filled with the 16bit TP value for stateful translation. High the TP value means high preference to perform translation.
- o IPv6_Prefix TLV has the type field assigned to 3. The length field is variable. The value field is filled with the IPv6 prefix for address mapping in stateless translation, encoding in NLRI format[RFC4760].
- o IPv4_pool TLV has the type field assigned to 4. The length field is variable. The value field is filled with the IPv4 pool for address mapping in stateful translation, encoding in NLRI format.

The AFBRs and BRs in the mesh should run MP-BGP process and peer with each other. When a new BGP session is established, AFBR and BR send a update containing the Translation Information Attribute to each other, which contains the Stateless_TP TLV or Stateful_TP TLV. Each router independently decides translation spot based on received TP

value. When the selected translation spot isn't the AFBR, then the AFBR should send another update with the Translation Information Attribute containing the IPv6_Prefix TLV or the IPv4_pool TLV to the BR. The tunnel-related routing should be triggered too, if there's any.

5. Further discussion

5.1. Achievement of translation spot selection

To be precise, through translation spot selection, we can solve the scalability problem of stateful translation and the operation complexity problem for both stateless and stateful translation. Also we make it more possible to perform application layer translation and adopt NAT46 mechanisms (NAT-PT) by pushing the translation spot down to the edge.

5.2. Cooperate with software

In the mesh scenario, software[RFC5565] is usually adopted as the tunnel mechanism. If it's used to support forwarding between the BR and the AFBR, then after translation spot signaling, BR and AFBR should trigger the software routing process, in which AFBR should advertise the actual IPv4 prefixes, while BR should advertise to AFBR either the address pool assigned from the AFBR (stateful case), or the IPv4 address prefix containing the IPv4 address possessed by the IPv6 hosts (stateless case).

5.3. Using NAT64 or IVI as translation mechanism

NAT64[I-D.ietf-behave-v6v4-xlate-stateful] is a typical stateful translation mechanism. It can be used in the IPv4/IPv6-coexist mesh for translation-involved communications across the backbone. If AFBR is chosen to be the translation spot, then the traffic will follow a traditional NAT64 process; else BR is chosen to be the translation spot, then AFBR should divided its public IPv4 address pool and assigned one block to the BR through translation spot signaling. BR will perform the NAT64 translation using the assigned IPv4 address block. In software routing, BR should advertise this block to AFBR.

IVI[I-D.xli-behave-ivi] is a typical stateless translation mechanism. It can be used in the IPv4/IPv6-coexist mesh for translation-involved communications across the backbone. If AFBR is chosen to be the translation spot, then the traffic will follow a traditional IVI process; else BR is chosen to be the translation spot, then AFBR should inform BR the IVI prefix, then BR can learn the address mapping role and the IPv4 prefix possessed by its network. In software routing, BR should advertise this IPv4 prefix to AFBR.

6. IANA considerations

IANA is requested to assign a value from the "BGP Path Attributes" Registry, to be called "Translation Information Attribute", with this document as the reference.

7. Acknowledgements

The authors would like to thank Lixia Zhang, Eric Nordmark, Jari Arkko, Alain Durand and David Ward for their valuable comments on this draft.

8. References

8.1. Normative References

- [RFC1702] Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation over IPv4 networks", RFC 1702, October 1994.
- [RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, March 1999.
- [RFC2765] Nordmark, E., "Stateless IP/ICMP Translation Algorithm (SIIT)", RFC 2765, February 2000.
- [RFC2766] Tsirtsis, G. and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)", RFC 2766, February 2000.
- [RFC2767] Tsuchiya, K., HIGUCHI, H., and Y. Atarashi, "Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)", RFC 2767, February 2000.
- [RFC2893] Gilligan, R. and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 2893, August 2000.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [RFC3338] Lee, S., Shin, M-K., Kim, Y-J., Nordmark, E., and A. Durand, "Dual Stack Hosts Using "Bump-in-the-API" (BIA)", RFC 3338, October 2002.
- [RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.
- [RFC4925] Li, X., Dawkins, S., Ward, D., and A. Durand, "Softwire Problem Statement", RFC 4925, July 2007.
- [RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, June 2009.

8.2. Informative References

[I-D.ietf-behave-v6v4-xlate-stateful]

Bagnulo, M., Matthews, P., and I. Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-v6v4-xlate-stateful-12 (work in progress), July 2010.

[I-D.xli-behave-ivi]

Li, X., Bao, C., Chen, M., Zhang, H., and J. Wu, "The CERNET IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition", draft-xli-behave-ivi-07 (work in progress), January 2010.

Authors' Addresses

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

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

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

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

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

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

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

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

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

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

Xing Li
Tsinghua University
Department of Electronic Engineering, Tsinghua University
Beijing 100084
P. R. China

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

Chris Metz
Cisco Systems, Inc.
3700 Cisco Way
San Jose, Ca. 95134
USA

Email: chmetz@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2011

R. Despres
RD-IPtech
October 18, 2010

IPv4 Residual Deployment across IPv6-Service networks (4rd)
A NAT-less solution
draft-despres-softwire-4rd-00

Abstract

During the long transition period from IPv4-only to IPv6-only, networks will have not only to deploy the IPv6 service but also to maintain some IPv4 connectivity for a number of customers, and this for both outgoing and incoming connections and for both customer-individual and shared IPv4 addresses. The 4rd solution (IPv4 Residual Deployment) is designed as a lightweight solution for this. It applies not only to ISPs have IPv6-only routing networks, but also to those that, during early transition stages, have IPv4-only routing, with 6rd to offer the IPv6 service, those that have dual-stack routing networks but with private IPv4 addresses assigned to customers.

In some scenarios, 4rd can dispense ISPs from supporting any NAT in their infrastructures. In some others it can be used in parallel with NAT-based solutions such as DS-lite and/or NAT64/DNS4 which achieve better IPv4-address sharing ratios (but at a price of significantly higher operational complexity).

Status of this Memo

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

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

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

This Internet-Draft will expire on April 21, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Definitions	4
3. Applicability	5
4. The 4rd Protocol Specification	8
4.1. Mapping Rules	8
4.2. Packet Encapsulations/Decapsulations	9
4.3. Port sets of IPv4r prefixes longer than /32	10
4.4. PMTU Considerations	12
4.5. Parameter Acquisitions by 4rd Clients	12
5. Example with IPv6-only Routing and Shared IPv4 Addresses	14
6. Security considerations	16
7. IANA Considerations	17
8. Acknowledgments	17
9. References	17
9.1. Normative References	17
9.2. Informative References	18
Author's Address	18

1. Introduction

During the long transition period from only IPv4 to IPv6, networks of Internet Service providers (ISPs) will have not only to offer IPv6 connectivity but also, for some customers, to maintain a residual IPv4 connectivity. Both outgoing and incoming connections will have to be supported. While some privileged customers will still have individual IPv4 addresses of their own, more and more others will only have shared IPv4 addresses.

All ISP routing networks will eventually be IPv6-only but, in earlier phases, some deployments of the IPv6 service can be done on ISP routing networks that only route private IPv4 of [RFC1918], the IPv6 service being offered by means of 6rd. Some others will route both IPv6 and private IPv4.

4rd is a solution for the residual support of global IPv4 connectivity across routing networks that are IPv6-only, private-IPv4-only, or IPv6-and-private-IPv4.

Depending on ISP constraints and policies, 4rd can be used across IPv6-only networks either alone, no NAT being then needed in ISP infrastructures, or in parallel with NAT based solutions that, at a price of more operational complexity, achieve better address sharing ratios such as [DS-lite] and [NAT64]/[DNS64].

This proposal is a more detailed version of what was initially described in section 3.2 of the more general Stateless Address Mapping proposal of [1]) (SAM).

At the time of writing, 4 ISPs in Japan have expressed interest for the SAM/4rd solution to offer IPv4 connectivity across IPv6-only routing networks (www.ietf.org/mail-archive/web/v6ops/current/msg05247).

2. Definitions

Locator: in a given address family, an address or a routable prefix.

IPv4r Address Family: the "residual IPv4" address family, that of IPv4r locators.

IPv4r Address: Either a global IPv4 address or the combination of a global IPv4 address and a port (an A+P address)

IPv4r prefix: Either a global IPv4 prefix (up to /32), or a global IPv4 address followed by a port-set identifier whose length is from 1 to 15.

IPv4p Address Family: That of a private address spaces of (10/8, 172.16/12, or 192.16/16, prefixes).

interior address family: in a tunnel-supporting network, the address family of encapsulating packets (in 4rd, IPv6 or IPv4p).

exterior address family: in a tunnel-supporting network, the address family of encapsulated packets (in 4rd, IPv4r).

4rd parent network: For a given 4rd network, the network that assigns to it one or several IPv4r prefixes.

4rd network: A network whose interior address family is different from global IPv4, and that supports one or several 4rd servers at its border with its 4rd parent network.

4rd server (4rd-S): A function at a border point between a 4rd network and its 4rd parent network. Via automatic tunnels, it statically shares among customers of the 4rd network IPv4r locators that have been received from the parent network.

4rd client (4rd-C): A function that obtains mapping rules from a 4rd server, derives from them its own IPv4r locator, and tunnels IPv4r packets across its 4rd network.

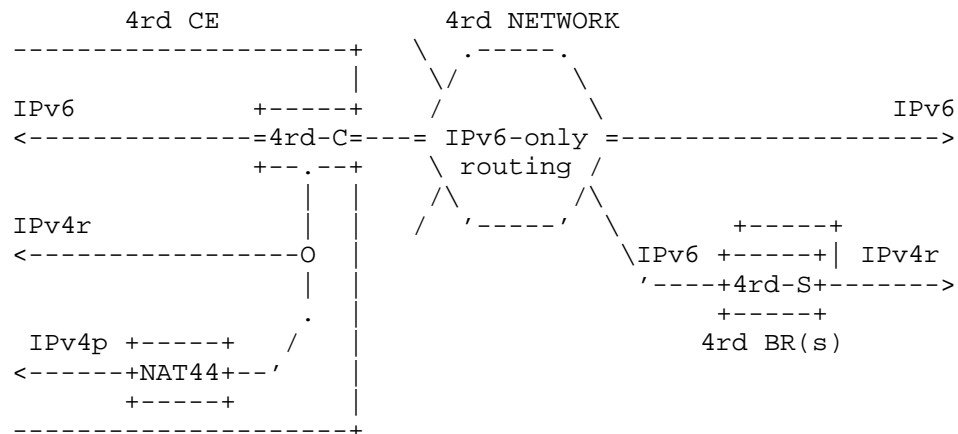
4rd BR: A router that supports one or several 4rd servers (Border Router).

4rd CE: A node supports a 4rd client and is in a customer position on a 4rd network. It may be a host, a router, or both.

network PMTU: For an identified address family, the packet size that must not be exceed to traverse the network without risk of packets being discarded (in IPv6) or fragmented (in IPv4).

3. Applicability

For 4rd to actually be used across a network, the network must be a 4rd network, and must have at least one 4rd CE.



4rd ACROSS AN IPv6-ONLY ROUTING NETWORK

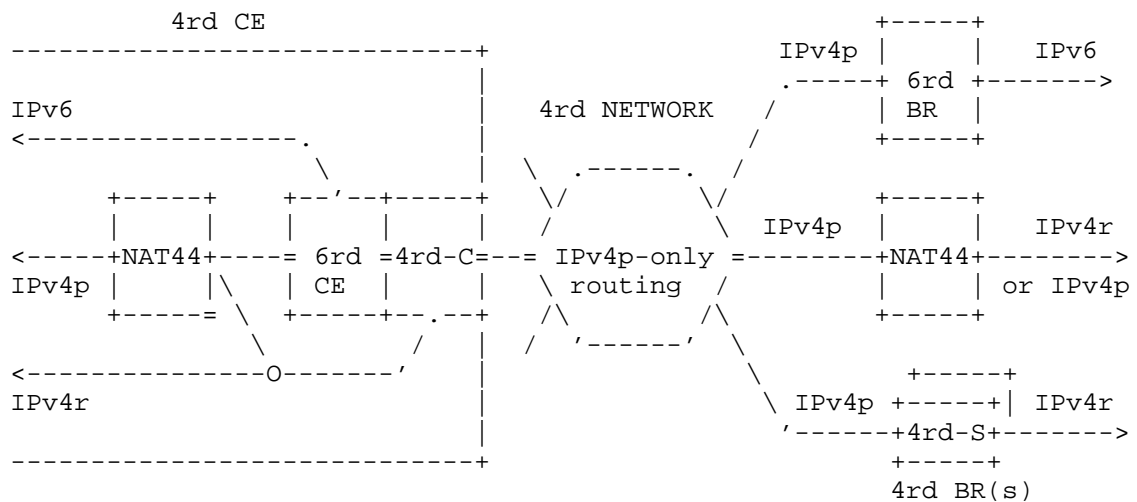
Figure 1

If the interior address family is IPv4p, the operator of must know the PMTU of its 4rd network.

Figure 1 shows a scenario where the interior address family is IPv6. In the CE, the IPv4r interface of the 4rd client can be used to provide global IPv4 addresses and reserved ports to a socket API and/or to a NAT44. This NAT can use them for its port-forwarding function, be it configured administratively or by means of UPnP or NAT-PMP. If both a socket API and a NAT44 share the set of available addresses and ports, a static switch can do split.

This scenario doesn't exclude other ways to offer IPv4 connectivity across the same IPv6-only routing network (typically DS-lite and/or NAT64/DNS64). Note however that, with each IPv4 address shared between 16 customers, each customer obtains with 4rd 3840 global-IPv4 ports (in addition to its 65 536 ports per IPv6 address), and the available IPv4 address space is multiplied by 16. Since most port-consuming applications should quickly be reachable in IPv6 (Google Maps in particular is already in this case) this should be largely sufficient in many scenarios.

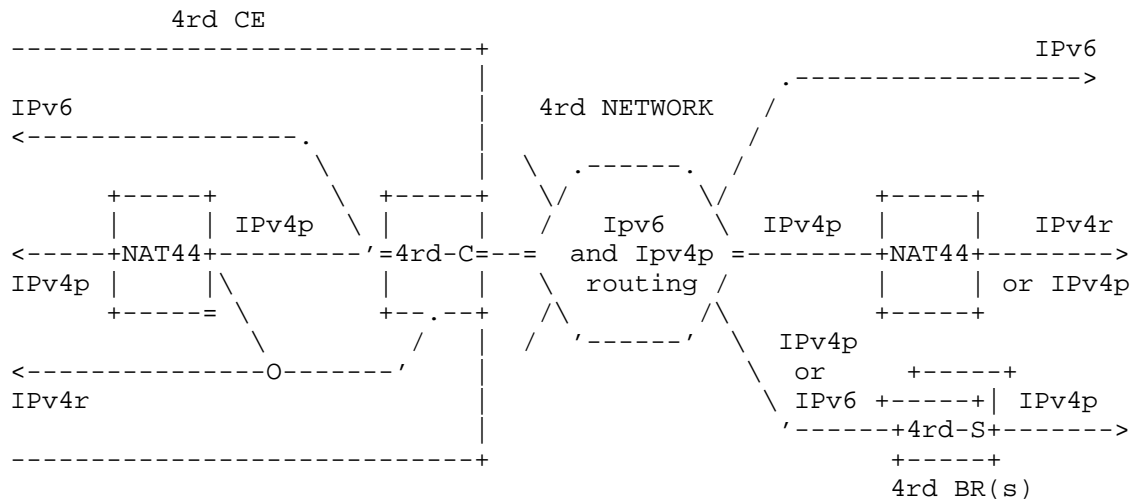
Figure 2 shows a scenario where the interior address family is IPv4p and where the IPv6 service is supported with 6rd. The 4rd CE architecture is similar to that of the previous example with two differences: IPv6, instead to be directly available at the network interface, is obtained by means of a 6rd-CE function; the NAT44, if present, can use as external addresses not only those of its IPv4r locator but also the IPv4 address assigned to the CE in the 4rd network. How the NAT44 uses this external address set is an implementation matter, but it can be noted that applications that are known to traverse cascades of NATs without problem (Web, DNS, and Mail, in particular) can use IPv4p addresses. IPv4r addresses are thus kept for IPv4 connections that may need end-to-end transparency.



4rd ACROSS AN IPv4p-ONLY ROUTING NETWORK

Figure 2

Figure 3 shows a scenario where both IPv6 and IPv4p are routed. The main difference with the IPv4p-only routing case is that 6rd is not needed. Tunnels for IPv4r packets can use IPv6 or IPv4p depending on local policies.



4rd ACROSS A DUAL-STACK ROUTING NETWORK

Figure 3

NOTE: The above scenarios can apply not only to 4rd networks operated to ISPs but also to private networks. A CPE that supports a 4rd server can, when it has an IPv4r locator, share it among hosts of its site that support 4rd clients. This is in practice a static alternative to UPnP and NAT-PMP for hosts to still have some IPv4 incoming connectivity.

4. The 4rd Protocol Specification

4.1. Mapping Rules

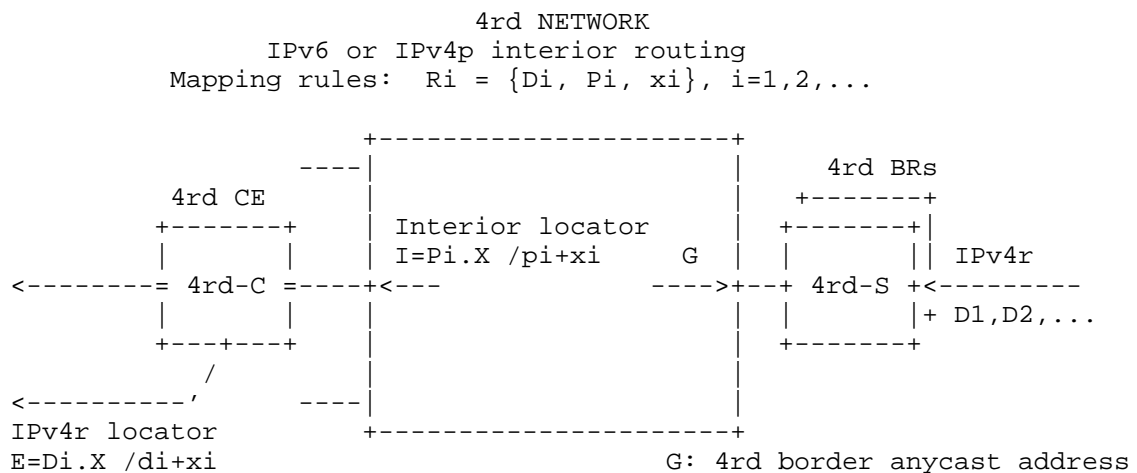
4rd mapping rules establish 1:1 mappings between interior and exterior locators. Each rule R_i comprises:

Di : the "rule exterior prefix"

Pi : the "rule interior prefix"

xi : the "index length", i.e. the length of the field X that, for a given 4rd client is common to its interior and exterior locators.

Di's of all rules of a 4rd network must be non overlapping prefixes,
and the same for Pi's.



4rd LOCATOR MAPPING RULES

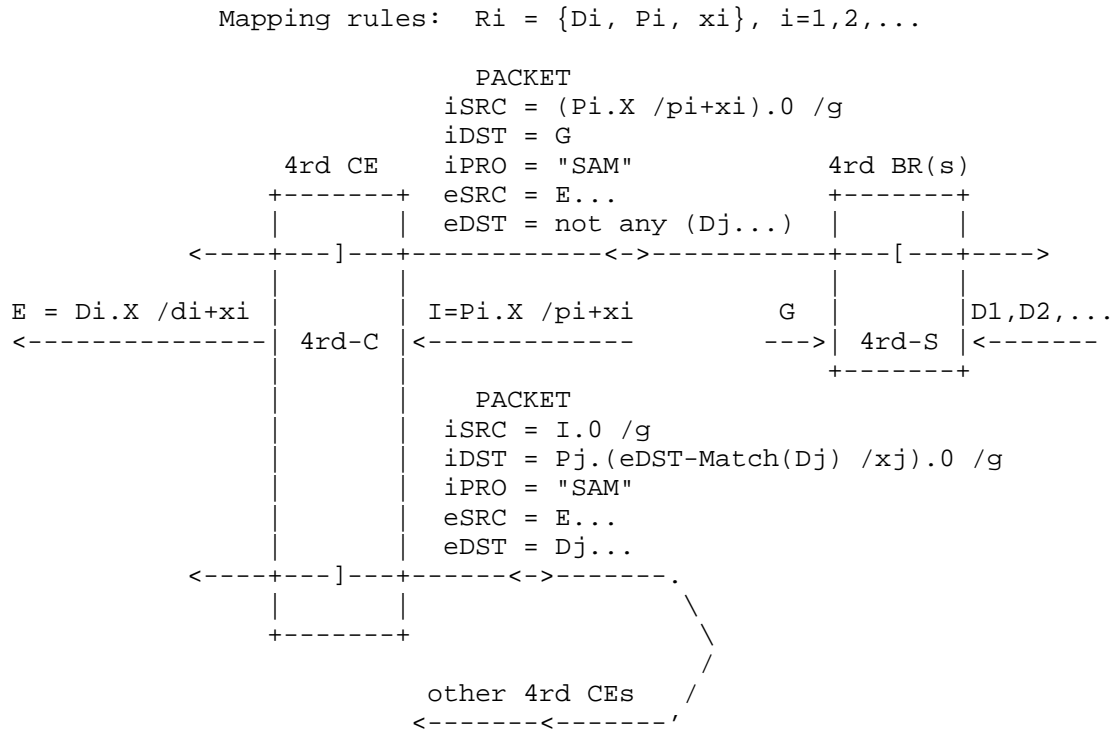
Figure 4

Figure 4 shows how the exterior locator "E" of a 4rd client is derived from its interior locator "I". E comprises the Di of the rule whose Pi is recognized at the beginning of I, followed by index X whose length is the xi of the rule, and which is copied from the I after the its Pi. In this document field acronyms are uppercase, and lengths of fields are the same letters in lower case. (Thus,

"/pi.xi" represents a locator length that is the sum of Pi's length and xi).

To derive an interior address from an exterior address, the reverse logic is used. In this document, Y... represents any address that starts with prefix Y. The interior locator I derived from exterior address E... then comprises the Pi of the rule whose Di matches the beginning of E..., followed by index X whose length is the xi of the rule and which is copied from E... after its bits used to match Di. If the obtained I is shorter than a complete interior address, it is completed with zeroes. If no rule applies (no Di found in E), the interior locator is the 4rd-server interior address G (an anycast address).

4.2. Packet Encapsulations/Decapsulations



4rd PACKET ENCAPSULATIONS AND ADDRESS MAPPINGS

Figure 5

When a 4rd client or server receives a packet at its IPv4r interface (a pseudo interface in the client case), it checks the validity of its source and destination addresses. It also checks that the packet size is acceptable (see Section 4.4). If yes, it encapsulates it in an interior packet and forwards it via its interior interface.

The Next-header field, if interior addresses are IPv6, of the Protocol field if they are IPv4p, a value to be assigned by IANA for 4rd and for other applications of the SAM of [1] (SAM). A specific value for SAM is preferred to a re-use of Protocol 41, used for IP-in-IP encapsulations of 6to4, ISATAP, and 6rd, because this ensures that coexistence with these without risk of incompatibility.

Symmetrically, a 4rd client or 4rd server that receives a packet at its interior interface checks the validity of source and destination addresses in both its encapsulating and encapsulated packets. It also checks that they are mutually consistent with mapping rules of the 4rd network. If yes it decapsulates the IPv4r packet contained in the encapsulating packet, and forwards it its IPv6 interface.

Details on which addresses are acceptable in which packets are detailed in Figure 5, where SRC and DST respectively mean source and destination, PRO means protocol, where iXXX and eXXX respectively refer to interior and exterior address families.

4.3. Port sets of IPv4r prefixes longer than /32

The port-set identifiers *S* of an IPv4r prefix of length *s* in the range 33 to 47 consists in the *s*-32 bits beyond the first 32. The port set it identifies is specified with the following constraints:

"Exclusiveness" Port sets of two *S*'s must be disjoint if the *S*'s are non overlapping prefixes (10 and 1011 do overlap while 10 and 1110 don't)

"No administration" The port set of *S* must be algorithmically derived from *S* without depending on any parameter.

"Fairness-1" Port sets of two *S*'s of same lengths must contain the same number of ports.

"Fairness-2" No port-set may contain any port 0 to 4095 (these have more value than others in OS's, and are normally not used in dynamic port assignments to applications).

4.4. PMTU Considerations

To properly deal with large size IPv4 datagrams that are fragmented before entering a 4rd network, precautions have to be taken because:

- o In IPv4, intermediate nodes may have to forward packets that are longer than the MTU of next links to be traversed. For this, they fragment packets within the network.
- o In IPv6, such packets are discarded, with ICMP Packet Too Big ICMPv6 error packets returned to sources, but with all IPv6 links having to support MTUs of at least 1280 octets.

To cope with these constraints, 4rd clients and 4rd servers can reassemble multi-fragment IPv4 datagrams before processing them. (This function is stateful at the IP layer like the same function in NATs. But at the transport layer, 4rd remains stateless whereas NATs are stateful, a source of operational complexity that is avoided with 4rd.)

Each datagram, after fragment reassembly if needed, is forwarded either in a single packet, if with its encapsulation header it fits in the network PMTU, or in as many packets as needed for each one to fit in this PMTU. Optimized treatments are possible, whereby first parts of datagrams are forwarded without waiting for complete datagram reassembly, but this is an implementation matter that doesn't belong to the scope of this specification.)

4.5. Parameter Acquisitions by 4rd Clients

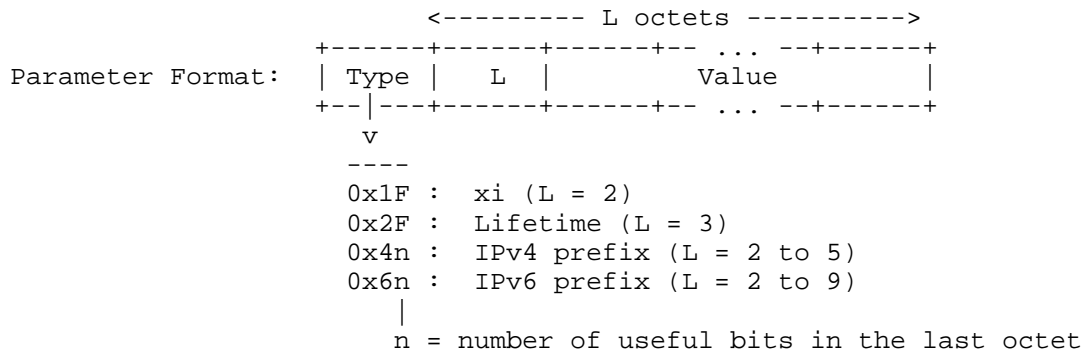
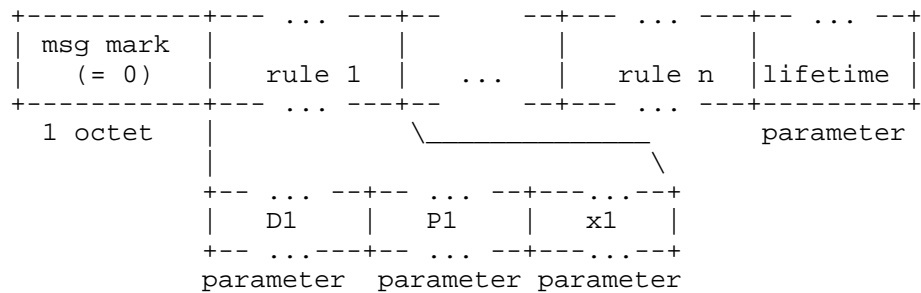
The 4rd-server address G may be obtained in various ways. It may be administratively configured (typically applicable if the 4rd network operator provides its own 4rd CEs). It can also be obtained in DHCP [RFC2131], DHCPv6 [RFC3315], Radius [RFC2865], or Diameter [RFC3588]. For these, IANA assigned numbers for 4rd remain to be chosen. In absence of all these means, G can be taken as the well-known address of SAM servers in the applicable interior address family (also to be assigned by IANA).

If a 4rd client has Gs for both IPv6 and IPv4p, it may try both and settle for either one from which it obtains responses.

To obtain its mapping rules and their common lifetime, a 4rd client sends a 4rd "Parameter Request" message to the 4rd-server anycast address G. It retransmits it until it obtains an answer, typically with longer time intervals after several unsuccessful attempts. When it receives a 4rd "Parameter Indication" message with the 4rd-server anycast address as source, it derives from the contained mapping

rules its own IPv4r locator. It also stores these rules for its future packet encapsulations/decapsulations.

4rd messages are transmitted in payloads of 4rd interior packets at the same place as encapsulated exterior packets. Their first octet is set to 0, a "Message Mark" which permits to distinguish 4rd messages from encapsulated packets (IPv4 packet headers all start with a 4 in the first 4 bits).



FORMAT OF 4rd PARAMETER INDICATIONS

Figure 7

A 4rd Parameter Request is sent with no information after the 4rd Message Mark. In order to facilitate future extensions that may prove useful, 4rd servers should ignore octets that may be received after this mark.

The following example illustrates the case of an ISP that operates an IPv6-only routing network and assigns shared global IPv4 addresses to its customers. The ISP has 2^{24} customers whose /48 prefixes start with a common prefix K/24. In IPv4, it has three global IPv4 prefixes, R1/13, R2/14, R3/14, giving a total of 2^{20} addresses. Each of these addresses must therefore be shared among 16 customers. Exterior locators E must therefore be /36s, comprising port-set identifiers S having 4 bits (each customer is thus assigned $2^{12} \times 15 / 6 = 3840$ reserved ports in global IPv4). Each interior prefix I/48 must then be composed of the common prefix K followed by the short identifier Ci of one of the three Di's. Their lengths have to be related to lengths of Di's by the formula $ci=(i-k)-(e-di)$, which gives $c1=1$, $c2=2$, and $c3=2$. Within these constraints, bit values of the Ci's may be arbitrary non overlapping prefixes, e.g. $C1 = 0b0$, $C2=0b10$, $C3 = 0b11$ (with 0bXXX being the binary number XXX). Rule are {D1/

VARIANTS:

- o It the ISP would have preferred to have only one rule, this would have been possible by using in IPv4 only the /13. Then port-set identifiers S would have had 5 bits, and each customer would have had 1920 ports in global IPv4.
- o If instead of one K/24, the ISP there would have had to use two different prefixes, K1/25 and K2/25, mapping rules could have been {D1/13, P1=K1/25, x1=23}, {D2/14, P2=K2.C2/26, x2=22}, and {D3/14, P3=K2.C2/26, x3=22}, with $C2=0b0$ and $C3=0b1$.
- o If, in a more complex scenario, the ratio between number of customers and number of IPv4 addresses would not have been a power of two, either some interior addresses or some exterior addresses would have had to be sacrificed (not assigned). For example, with K1/25, K2/26, and D1/14, D2/15, D3/15, D4/15, giving $2^{23} + 2^{22}$ customers and $2^{19} + 2^{15}$ IPv4 addresses, rules could have been {D1/14, P1=K1.C1/26, x1=22}, {D2/15, P2=K1.C2/27, x2=21}, {D3/15, P3=K1.C3/27, x3=21}, {D4/15, P4=K2.C4, x4=21}, with $C1=0b0$, $C2=0b10$, $C3=0b11$, $C4=0$.

6. Security considerations

Spoofing attacks

With address-consistency checks of Section 4.2, authentication verifications that apply interior locators also apply, indirectly, to exterior locators. Similarly, anti-spoofing protections that apply to interior addresses also apply, indirectly, to exterior locators. 4rd should therefore introduce no opportunity of its own for spoofing attacks.

Denial-of-service attacks

Reassembly of fragmented exterior datagrams introduces an opportunity for some form of DOS attacks, shared with NAT-based solutions. Note that this risk among reason to prefer native IPv6 to native IPv4 when there is the choice for a transport connection.

Risks of DOS attacks at the transport-connection layer, to which NAT-based solutions are exposed, are avoided in 4rd because of its the stateless operation of this layer.

Faked 4rd servers

If a 4rd CE uses as 4rd server address one of the two IANA assigned well-known address for this in IPv6 and IPv4, and if its ISP network has no 4rd server, packets addressed to it can be forwarded to the Internet backbone. They should however not reach any faked 4rd server because, this address starting with none of prefixes routed to other ISP networks, they will normally be discarded in the backbone. However, whether some additional protection in would be appropriate against fake 4rd servers (e.g. with a nonce in Parameter Requests and Parameter Indications), is still viewed as an open issue.

Routing-loop attacks

Routing-loop attacks that may exist in some automatic-tunneling scenarios are documented in [3]. They cannot exist with 4rd because its address checks of Section 4.2 prevent multiple traversals of a 4rd network by the same IPv4r packet, and because, 4rd using its own Protocol number, routing-loops between nodes of nodes working with two different tunnel protocols are also impossible.

7. IANA Considerations

This specification depends on the following number assignments by IANA:

- o The SAM protocol number (Section 4.2)
- o The DHCP and DHCPv6 4rd option codes (Section 4.5)
- o The Radius 4rd attribute type (Section 4.5)
- o The SAM-server well-known addresses, in IPv4 and IPv6 (Section 4.5)

8. Acknowledgments

The author has benefited from useful informal discussions with a number of IETF participants on previous SAM proposals, from which this specification is a by-product. Concerning 4rd in particular, Satoru Matsushima deserves special recognition, first for the interest in the approach he expressed from the beginning, but also for his constructive contributions, including his proposal of the 4rd acronym, and for convincing his colleagues to make actual deployment plans with this technology. Olivier Vautrin, by independently proposing the same acronym for a similar orientation, has to be thanked for the valuable encouragement this has been.

9. References

9.1. Normative References

- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [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.

- [RFC3513] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.
- [RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

9.2. Informative References

- [1] Despres, R., "Stateless Address Mapping (SAM) - a Simplified Mesh-Software Model - draft-despres-software-sam-01 - work in progress", July 2010.
- [2] Vautrin, O., "IPv4 Rapid Deployment on IPv6 Infrastructures (4rd) - draft-vautrin-software-4rd-00 - work in progress", July 2010.
- [3] Nakibly, G. and F. Templin, "Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations - draft-ietf-v6ops-tunnel-loops-00 - Work in progress", September 2010.
- [DNS64] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers [draft-ietf-behave-dns64 - work in progress]", October 2010.
- [DS-lite] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion [draft-ietf-software-dual-stack-lite - work in progress]", August 2010.
- [NAT64] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 - Clients to IPv4 Servers [draft-ietf-behave-v6v4-xlate-stateful - work in progress]", July 2010.

Author's Address

Remi Despres
RD-IPtech
3 rue du President Wilson
Levallois,
France

Email: remi.despres@free.fr

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 15, 2011

R. Despres
RD-IPtech
B. Carpenter
Univ. of Auckland
S. Jiang
Huawei Technologies Co., Ltd
October 12, 2010

Native IPv6 Across NAT44 CPEs (6a44)
draft-despres-softwire-6a44-01

Abstract

Most CPEs should soon be dual stack, but a large installed base of IPv4-only CPEs is likely to remain for several years. Also, with the IPv4 address shortage, more and more ISPs will assign private IPv4 addresses to their customers. The need for IPv6 connectivity therefore concerns hosts behind IPv4-only CPEs, including such CPEs that are assigned private addresses. The 6a44 mechanism specified in this document addresses this need, without limitations and operational complexities of Tunnel Brokers and Teredo to do the same.

6a44 is based on an address mapping and on a mechanism whereby suitably upgraded hosts behind a NAT may obtain IPv6 connectivity via a stateless 6a44 server function operated by their Internet Service Provider. With it, IPv6 traffic between two 6a44 hosts in a single site remains within the site. Except for IANA numbers that remain to be assigned, the specification is intended to be complete enough for running codes to be independently written and interwork.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 15, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Applicability	4
3. 6a44 IPv6 Address Format	6
4. Address Mappings and Encapsulations	8
5. MTU considerations	10
6. Host Acquisition of IPv6 Addresses and their Lifetimes	10
7. Security considerations	13
8. IANA Considerations	14
9. Acknowledgments	14
10. References	14
10.1. Normative References	14
10.2. Informative References	15
Authors' Addresses	16

1. Introduction

Most CPEs (customer premise equipments) should soon be dual stack, but a large installed base of IPv4-only CPEs is likely to remain for several years. Also, with the IPv4 address shortage, more and more Internet service providers (ISPs) will assign private IPv4 addresses of [RFC1918] to their customers. The need for IPv6 connectivity therefore includes hosts behind IPv4-only CPEs, including such CPEs that have private addresses.

At the moment, there are two traversal techniques to address this need:

1. A configured tunnel (IPv6 in IPv4 or even IPv6 in UDP), involving a managed tunnel broker, e.g. [RFC3053], with which the user must register. Well known examples include deployments of the Hexago tool, and the SixXs collaboration. However, this approach does not scale well; it requires significant support effort and is really only suitable for "hobbyist" early adopters of IPv6.
2. Teredo [RFC4380]. This is an automatic UDP-based tunneling solution that relies on a Teredo server, and on Teredo relays willing to carry the traffic. Unfortunately experience shows that this is sometimes an unreliable process in practice, with clients sometimes believing that they have Teredo connectivity when in fact they don't, or alternatively with the Teredo server and relay being very remote from the client and causing extremely long latency for IPv6 packets. This leads to user frustration and even to advice from help desks to disable IPv6.

6a44 is based on an address mapping and on a mechanism whereby suitably upgraded hosts behind a NAT may obtain IPv6 connectivity via a stateless 6a44 server function operated by their Internet Service Provider.

To address this need without the mentioned limitations, 6a44 is based on an address mapping and on a mechanism whereby suitably upgraded hosts behind a NAT may obtain IPv6 connectivity via a stateless 6a44 server function operated by their ISP. It can apply even with ISPs that, due to the IPv4 address shortage, assign private addresses of [RFC1918] to their IPv4 customers (typically with prefix 10.0.0.0/8).

6a44 is only a transition technology. It will no longer have to be used when the number of IPv4-only CPEs becomes negligible.

Except for IANA numbers that remain to be assigned, the specification is intended to be complete enough for running codes to be independently written and interwork.

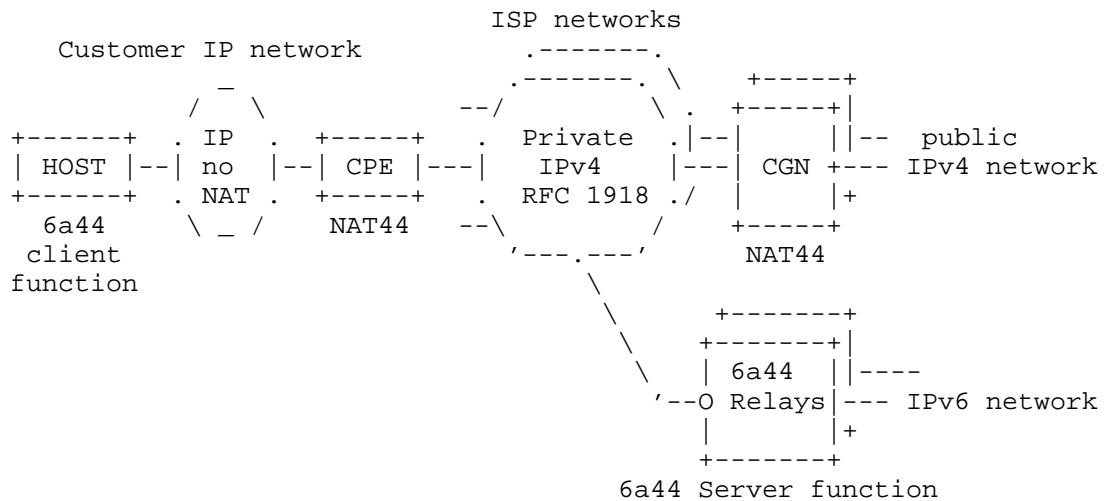
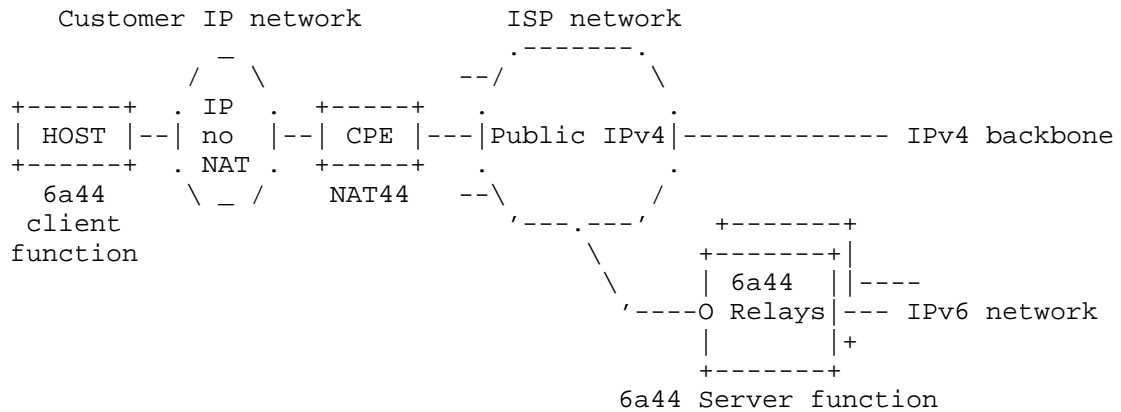
2. Applicability

Both hosts and ISPs can be made 6a44 capable independently of each other, with 6a44 being actually used by 6a44 capable hosts where their local ISPs are 6a44-capable.

For a host to be 6a44 capable, it has to support the 6a44 Client function ("6a44-C" in some Figures). This function is placed in its TCP/IP stack at the same place as the 6to4 router function of [RFC3056]: it has an IPv4 interface in its link-layer direction and both an IPv4 interface and an IPv6 pseudo-interface in its higher layers direction.

To enable its 6a44 function, a host must have no intra-site NAT44 between itself and the site CPE. (In sites where there are intra-site NAT44s, these NATs should be configured so that hosts behind them cannot enable 6a44. In view of the specification below, it can be done with a port mapping in each of them between the well-known port of 6a44 and an internal private address that DHCP doesn't assign.) In addition, the host must have in IPv4 a link MTU of at least 1308 octets (the MTU to be guaranteed in IPv6 plus the length of an UDP/IPv4 encapsulation header).

For an IPv4 ISP network to be 6a44 capable, the ISP must operate the 6a44 Server function, ("6a44-S" in some Figures). This function is anywhere at its border between the IPv4 network and an IPv6 network in which it has a /48 prefix. Typically this prefix will be chosen from whatever shorter PA prefix has been allocated to the ISP. The 6a44 server function can be replicated in any number of routers, known as "6a44 Relays", to enhance service quality and service availability. Also, the network must have an IPv4 MTU of at least 1308 octets and, for security, must support the ingress filtering of [RFC3704] (see Section 7).



6a44 ISP CONFIGURATIONS

Figure 1

Each ISP can support one public-addressing and several private-addressing 6a44 networks.

In 6a44 networks, ISPs may route IPv6 in addition to IPv4. Where this is the case, 6a44 only concerns CPEs that are IPv4-only capable. If on the contrary IPv4 is the only routed address family, 6a44 may also concerns sites where CPEs are dual-stack capable. Unable to take advantage of their IPv6 capability, they act as if they would be IPv4-only.

Figure 1 illustrates ISP-network configurations on which 6a44 can be used.

NOTE: The objective of 6a44 differs from that of Teredo ([RFC4380] and [RFC5991]). Teredo has been designed to avoid needing any ISP participation. This has permitted early deployment but didn't ensure connectivity between all Teredo addresses and all native IPv6 addresses. Also, it imposed a very significant level of complexity. On the contrary, 6a44 is designed to be explicitly supported by ISPs. As a result, connectivity between 6a44 IPv6 addresses and all native IPv6 addresses can be ensured, and implementations can remain simple.

3. 6a44 IPv6 Address Format

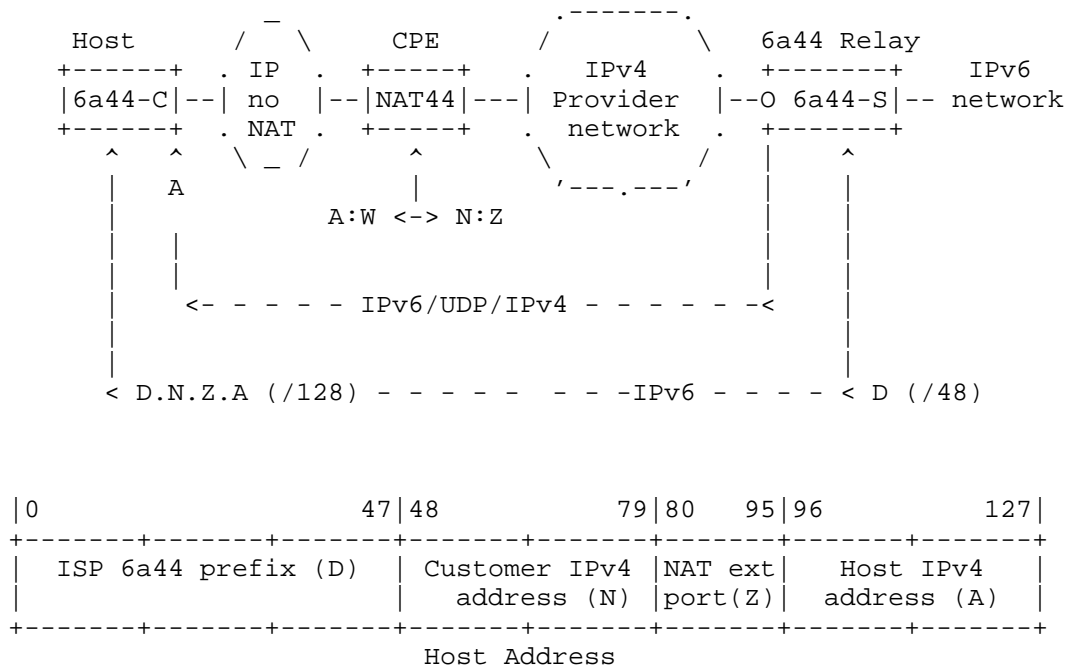


Figure 2

The 6a44 IPv6 address an ISP assigns to a host must first contain all what is needed to reach it from the IPv6 backbone. This includes, as illustrated in Figure 2:

- o the IPv6 prefix D that the ISP has assigned border routers of its 6a44 network;
- o the IPv4 address N of the customer site (external address of the NAT44 in its CPE);
- o the port Z that, in the CPE NAT44 CPE, has to be used to reach the host at its address address A, and in the host the 6a44 well-known port W (to be assigned by IANA).

To ensure that two 6a44 hosts behind the same IPv4-only CPE exchange packets without entering the ISP network, the 6a44 address of each host must also contain its IPv4 address A.

The format of 6a44 IPv6 addresses, a concatenation of D,N,Z, and A, where D has to be a /48 prefix, is detailed in Figure 2.

NOTE: Since IPv6 prefixes D assigned by ISPs to their customers always start with 001, the prefix of all IPv6 Aggregatable Global Unicast addresses specified in [RFC2374], 6a44 IPv6 addresses bend the rule of [RFC4291] that says 'for all unicast addresses, except those that start with binary value 000, Interface IDs are required to be 64 bits long and to be constructed in Modified EUI-64 format'. This is however acceptable in practice because 6a44 addresses are never used on any real IPv6 link, and in particular never subject to the neighbor discovery protocol of [RFC2461] which depends on properties of interface IDs. A revision of the [RFC4291] sentence should eventually clarify this point.

4. Address Mappings and Encapsulations

Figure 3 and Figure 4 detail the address mappings and encapsulations/decapsulations to be performed by 6a44 Client and server functions respectively, with the following notation:

- o (vX,A1,A2,data): a packet of the IPvX version that has A1 as source address, A2 as destination address, and "data" as payload.
(UDP,P1,P2,data): a UDP IP payload that has P1 as source port, P2 as destination port, and "data" as payload.
- o B is the 6a44 well-known anycast address, that of the 6a44 Server function. X...: an address that starts with prefix X.
- o not X: an address different from X
- o X.Y: the concatenation of X and Y (the dot is the concatenation operator).

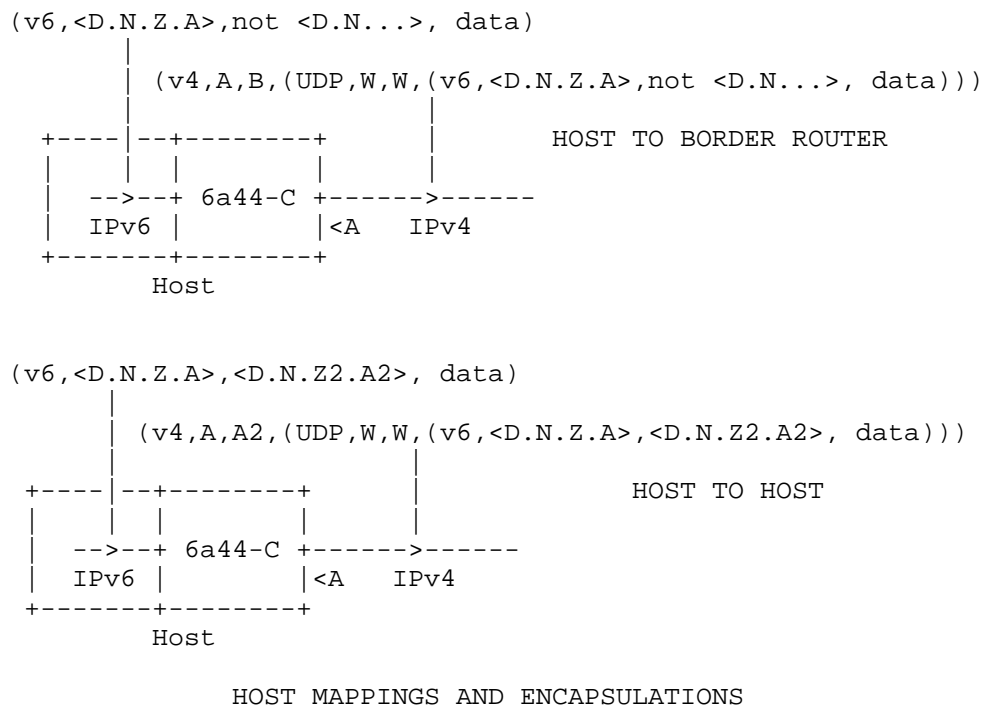


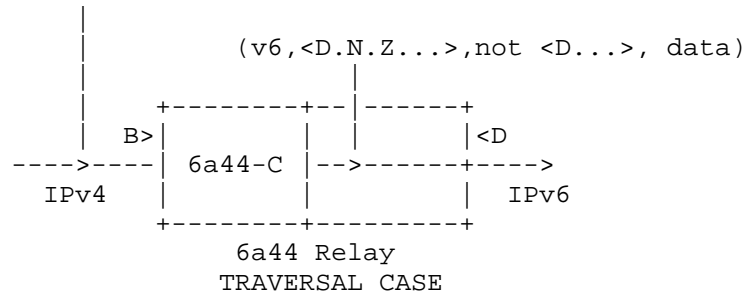
Figure 3

For protection against spoofing attacks, decapsulating functions must check consistency of IPv6 addresses fields with IPv4 addresses and UDP ports of encapsulating headers, both for source and destination addresses.

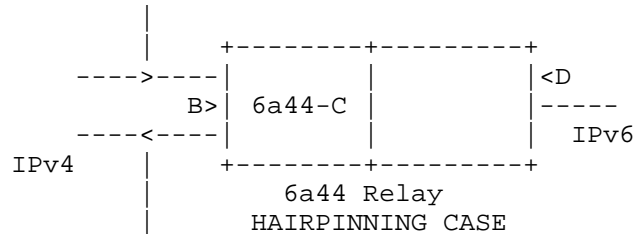
Figures present only one direction of 6a44-function traversals, but mappings that apply to the reverse direction are the same, with just a permutation of source and destination fields, for all of IPv4, IPv6, and UDP. Mappings and encapsulations/decapsulations for the reverse direction of that presented in Figures are the same, but with source and destination permuted in IPv6, IPv4 and UDP.

Recommendations of [RFC4213] that concern these encapsulations have to be followed.

$(v4, \langle N = \text{not } B \rangle, B, (\text{UDP}, Z, W, (v6, \langle D.N.Z \dots \rangle, \text{not } \langle D \dots \rangle, \text{data})))$



$(v4, \langle N1 = \text{not } B \rangle, B, (\text{UDP}, Z1, W, (v6, \langle D.N1.Z1 \dots \rangle, \langle D.N2.Z2 \dots \rangle, \text{data})))$



$(v4, B, N2, (\text{UDP}, B, Z2, (v6, \langle D.N1.Z1 \dots \rangle, \langle D.N2.Z2 \dots \rangle, \text{data})))$

6a44-RELAY MAPPINGS AND ENCAPSULATIONS

Figure 4

5. MTU considerations

Reassembly of multi-fragment datagrams needs stateful processing, and opens the door to some denial of service attacks. To ensure a freedom of distribution of 6a44 Server functions in any number of parallel processors anywhere in 6a44 ISP networks, it has therefore to be avoided.

For this:

- o 6a44 ISP networks must have internal IPv4 MTUs of at least 1308 octets (which is easy to ensure).
- o 6a44 hosts must limit to 1280 octets IPv6 packets they transmit to destinations that are not neighbors on their own links. This behavior is already the normal one as long as no other IPv6 path MTU has been reliably discovered.
- o 6a44 Server functions refuse packets received from their IPv6 pseudo interfaces if their sizes exceed 1280 octets, with ICMPv6 Packet Too Big messages returned to sources as required by [RFC2460].)

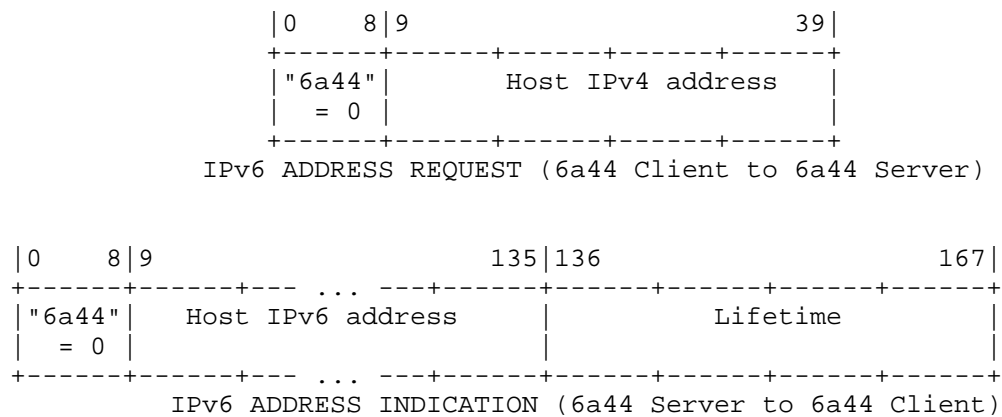
In a host, a destination is considered to be an on link neighbor if the IPv6 destination has the same bits 0-79 as the host address, and if the IPv4 destination starts with the prefix of the IPv4 link of the host. In this case, the IPv6 path MTU can be taken as that of the IPv4 link MTU minus 28 octets (a value that is typically significantly longer than 1280 octets).

6. Host Acquisition of IPv6 Addresses and their Lifetimes

Acquisition of 6a44 addresses by hosts is independent from other mechanisms they may have to acquire other IPv6 addresses (PPP, DHCP, SLAAC, ...). It only depends on 6a44 packet exchanges between hosts and 6a44 Relays.

In order to acquire 6a44 addresses, hosts transmit IPv6 Address Request messages to 6a44 Server functions and expect IPv6 Address Indication messages in return.

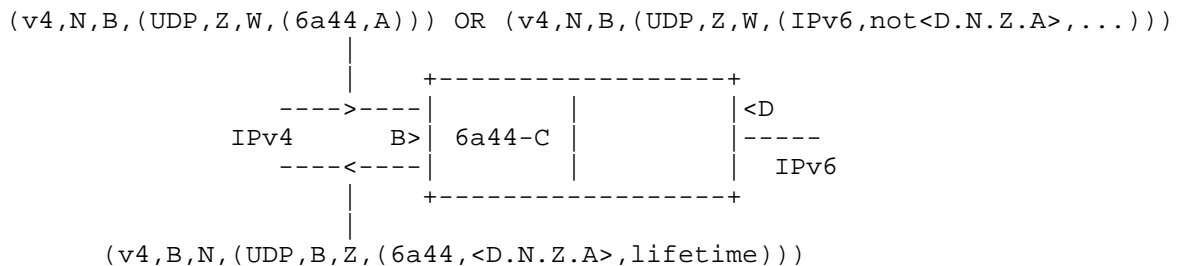
Formats of these 6a44 messages are shown in Figure 5. They start with a 6a44 mark, a null octet chosen so that, in payloads of UDP datagrams received by 6a44 Client and 6a44 Server functions, 6a44 messages can be distinguished from IPv6 packets (IPv6 packets always have a non-null first octet).



6a44 MESSAGES

Figure 5

Message processing in 6a44 Server function is shown in Figure 6 with the same notation as in Section 4. The lifetime of returned IPv6 addresses should be the same as that of IPv4 addresses assigned by the same ISP. it is expressed in seconds.



6a44 MESSAGE PROCESSING IN BORDER ROUTERS

Figure 6

In a host, the 6a44 Client function should be activated for one of its physical interfaces only if this interface has a private IPv4 address and no other native IPv6 address. (An address is said to be native if it starts with 2000::/3 (global unicast) and neither with 2002::/16 (the 6to4 prefix) nor with 2001::/32 (the Teredo prefix).)

Message processing in a 6a44 Client function consists in transmitting from time to time IPv6 Address Requests to the 6a44 Server function, and to update the host IPv6 address and its lifetime each time an IPv6 Address Indication message is received (with due IPv4 source address verification for security).

In order to decide when to transmit such a message, the 6a44 Client function has the equivalent of the following states:

"Waiting for an IPv6 Address Indication": When this state is entered, an IPv6 Address Request is transmitted, a Response Awaited timer of 1 second is started, and a Retransmission Count is set to 0. If the timer expires with a Retransmission Count less than 10, a new IPv6 Address Request is transmitted, and the count is increased by 1. If it expires with a count equal to 10, the state is changed to "waiting before a new attempt to find a 6a44 service". If an IPv6 Address Indication is received while in this state, the timer is stopped, the state is changed to "Waiting for having to refresh the NAT-binding". This state is also re-entered each time a new IPv4 address is assigned to the link-direction interface of the 6a44 Client function.

"Waiting for having to refresh the NAT-binding": When this state is entered, a timer of 29 second is started. (This value is that chosen for SIP in [RFC5626] for the same objective, i.e. to maintain tunnel NAT bindings without particular knowledge about NAT specifics.) This timer is restarted each time an IPv6 packet is transmitted to the 6a44 Server function (not when a packet is transmitted host to host within the customer site). It is also restarted if an IPv6 Address Indication is received while in this state. (This may happen in particular if the NAT binding has changed, e.g. because CPE reset during the lifetime of the IPv6 address.) If the timer expires, the state is changed to "Waiting for an IPv6 Address Indication".

"waiting before a new attempt to find a 6a44 service": When this state is entered, a 6a44 Availability timer of 1 hour is started. When it expires, the state is changed to "Waiting for an IPv6 Address Indication".

7. Security considerations

Traffic-capture attack by a neighbor: If it would be possible to transmit from a neighboring site a bogus address indication to a 6a44 host, this host could inadvertently advertise an IPv6 address that is not his real 6a44 address. Some incoming connections that it should have received could then be redirected to a wrong address. However, because 6a44 is applicable only to ISP networks that support the ingress filtering of [RFC3704] (see Section 2), no neighbor can fake a valid Address Indication message (the IPv4 source of packets it sends cannot be the 6a44 well-known IPv4 address, the only valid source for an Address Indication message).

Spoofing attacks: With address checks of Section 4, 6a44 should introduce no spoofing vulnerabilities beyond those the underlying IPv4 networks may have. ISPs that use subscriber authentications to secure IPv4 address assignments have the effect of this authentication automatically extended to 6a44 addresses (they include the assigned IPv4 addresses).

Denial-of-service attacks: Provided 6a44 Server functions are provisioned with enough processing power, which is facilitated by their being stateless, 6a44 is expected to introduce no denial of service vulnerabilities of its own.

Subscriber authentication: This is not provided as part of 6a44, because it is assumed to have occurred when the IPv4 address assignment was made.

Routing-loop attacks: A risk of routing-loop attacks has been identified for some encapsulation/decapsulation mechanisms [draft-ietf-v6ops-tunnel-loops-00]. It doesn't exist with 6a44 because:

- * IPv4 packets entering a 6a44 Server function are not forwarded if they come from another instance of the 6a44 Server function itself, i.e. if the IPv4 source is the 6a44 well-known IPv4 address Section 4.
- * The encapsulation header, which is based on UDP with a specific well-known port, cannot be confused with that of other encapsulation mechanisms (in particular those of IP in IP like those of 6to4, 6rd and ISATAP).

Missing 6a44 Server function: If a 6a44-capable host is client of an ISP that doesn't support 6a44, 6a44 IPv6 Address Request messages transmitted by the host will be forwarded to the Internet backbone, with the 6a44 well-known IPv4 address as destination. Since this address doesn't start with any prefix that the backbone routes toward ISP networks, these messages will be discarded before reaching any place where a fake 6a44 Server could have been malevolently placed. There is therefore no danger that 6a44 hosts could have their IPv6 traffic routed via 6a44 Server functions that would not belong to their local ISP (i.e. where they could be observed and acted upon without control).

8. IANA Considerations

For 6a44 to be used, both its IPv4 well-known address B and its well-known port W need to be assigned by IANA.

This assignment is necessary to validate the plug-an-play operation of 6a44 with independent implementations. Having it as quickly as possible (i.e. without waiting for all details of the specification to be agreed on), would be helpful for an early validation of the 6a44 plug-and-play operation.

9. Acknowledgments

This specification results from a convergence effort of authors of [draft-despres-softwire-6rdplus-00] and [draft-carpenter-softwire-sample-00]. Useful comments have been received about these earlier proposals or later, in particular from Pascal Thubert, Dan Wing, Yu Lee, Olivier Vautrin, Fred Templin, and Ole Troan. They have to be thanked for their contributions.

10. References

10.1. Normative References

- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

10.2. Informative References

- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2374] Hinden, R. and S. Deering, "An IPv6 Aggregatable Global Unicast Address Format", RFC 2374, July 1998.
- [RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.
- [RFC3053] Durand, A., Fasano, P., Guardini, I., and D. Lento, "IPv6 Tunnel Broker", RFC 3053, January 2001.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.
- [RFC5626] Jennings, C., Mahy, R., and F. Audet, "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC 5626, October 2009.
- [RFC5991] Thaler, D., Krishnan, S., and J. Hoagland, "Teredo Security Updates", RFC 5991, September 2010.
- [draft-carpenter-softwire-sample-00]
Carpenter, B. and S. Jiang, "Legacy NAT Traversal for IPv6: Simple Address Mapping for Premises - Legacy Equipment (SAMPLE)", June 2010.
- [draft-despres-softwire-6rdplus-00]
Despres, R., "Rapid Deployment of Native IPv6 Behind IPv4 NATs (6rd+)", July 2010.
- [draft-ietf-v6ops-tunnel-loops-00]
Nakibly, G. and F. Templin, "Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations - Work in progress", September 2010.

Authors' Addresses

Remi Despres
RD-IPtech
3 rue du President Wilson
Levallois,
France

Email: remi.despres@free.fr

Brian Carpenter
Department of Computer Science
University of Auckland
PB 92019
Auckland, 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Sheng Jiang
Huawei Technologies Co., Ltd
KuiKe Building, No.9 Xinxu Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing,
P.R. China

Email: shengjiang@huawei.com

Network Working Group
Internet Draft
Intended status: Standards Track

Dayong Guo
Sheng Jiang
Huawei Technologies Co., Ltd
R. Despres
RD-IPtech
Oct 18, 2010

Expires: April 25, 2011

RADIUS Attribute for 6rd

draft-guo-software-6rd-radius-attr-00.txt

Status of this Memo

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

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

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

This Internet-Draft will expire on April 25, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Abstract

6rd is One of the most popular methods to provide both IPv4 and IPv6 connectivity services simultaneously during the IPv4/IPv6 co-existing period. The DHCP 6rd option has been defined to configure 6rd CPE. But in many networks, the configuration information may be stored in AAA servers while user configuration is mainly from Broadband Network Gateway (BNG) through DHC protocol. This document defines a RADIUS attribute that carries 6rd configuration information from AAA server to BNG.

Table of Contents

1. Introduction.....	3
2. Terminology.....	3
3. 6rd Configuration with RADIUS.....	3
4. Attributes.....	4
4.1. 6rd Attribute.....	4
4.2. Table of attributes.....	5
5. Diameter Considerations.....	6
6. Security Considerations.....	6
7. IANA Considerations.....	6
8. Acknowledgments.....	6
9. Change Log [RFC Editor please remove].....	6
10. References.....	7
10.1. Normative References.....	7
10.2. Informative References.....	7

1. Introduction

Recently providers start to deploy IPv6 and consider how to transit to IPv6. 6rd [RFC5969] is one of the most popular methods to provide both IPv4 and IPv6 connectivity services simultaneously during the IPv4/IPv6 co-existing period. 6rd is used to provide IPv6 connectivity service through legacy IPv4-only infrastructure. 6rd adopt DHCP as auto-configuring protocol. The 6rd CPE extends DHCP option to discover 6rd border relay and to configure IPv6 prefix and address.

In many networks, user configuration information may be managed by AAA servers, together with user Authentication, Authorization, and Accounting (AAA). Current AAA servers communicate using the RADIUS (Remote Authentication Dial In User Service, [RFC2865]) protocol. In a fixed line broadband network, the Broadband Network Gateways (BNGs) act as the access gateway of users (hosts or CPEs). The BNGs are assumed to embed a DHCP server function that allows them to locally handle any DHCP requests issued by hosts.

Since the 6rd configuration information is stored in AAA servers and user configuration is mainly through DHC protocol between BNGs and hosts. New RADIUS attributes are needed to propagate the information from AAA servers to BNGs.

2. Terminology

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

3. 6rd Configuration with RADIUS

The below Figure 1 illustrates how the RADIUS protocol and DHCP are cooperated to provide users/hosts with 6rd configuration.

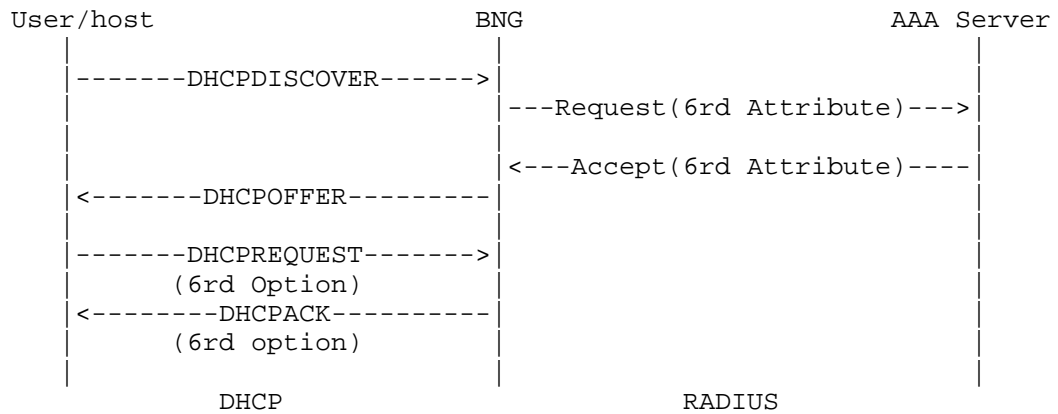


Figure 1: the cooperation between DHCP and RADIUS

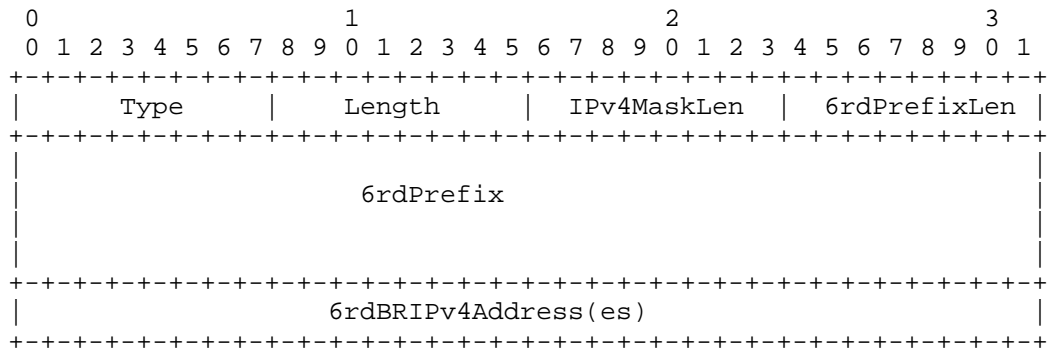
BNGs act as a bridge between user and AAA server. First, a BNG receives a user DHCPDISCOVER. It initiates the BNG to request correspondent user authentication relevant from an AAA server using RADIUS protocol. A 6rd request may be also sent in the same message. If the user authentication is approved by the AAA server, an Accept message is acknowledged with the 6rd Attribute, defined in the next Section. After the BNG responds to the user with an Advertise message, the user requests for a 6rd Option. Then, the BNG can reply the user using DHCP protocol.

4. Attributes

This section defines 6rd attribute which is used in the 6rd scenario.

4.1. 6rd Attribute

The 6rd Attribute is structured as follows:



Type TBD

Length the length of the DHCP option in octets (22 octets with one BR IPv4 address).

IPv4MaskLen The number of high-order bits that are identical across all CE IPv4 addresses within a given 6rd domain. This may be any value between 0 and 32. Any value greater than 32 is invalid.

6rdPrefixLen The IPv6 Prefix length of the Service Provider's 6rd IPv6 prefix in number of bits. The 6rdPrefixLen MUST be less than or equal to 128.

6rdPrefix The Service Provider's 6rd IPv6 prefix represented as a 16 octet IPv6 address. The bits after the 6rdPrefixlen number of bits in the prefix SHOULD be set to zero.

6rdBRIPv4Address One or more IPv4 addresses of the 6rd Border Relay(s) for a given 6rd domain.

4.2. 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	#	Attribute
				Request		
0+	0+	0	0	0+	TBD	6rd

5. Diameter Considerations

This attribute is usable within either RADIUS or Diameter [RFC3588]. Since the Attributes defined in this document will be allocated from the standard RADIUS type space, no special handling is required by Diameter entities.

6. Security Considerations

In 6rd scenarios, the RADIUS protocol is run over IPv4. Known security vulnerabilities of the RADIUS protocol are discussed in RFC 2607 [RFC2607], RFC 2865 [RFC2865], and RFC 2869 [RFC2869]. Use of IPsec [RFC4301] for providing security when RADIUS is carried in IPv6 is discussed in RFC 3162 [RFC3162].

Security considerations for the Diameter protocol are discussed in RFC 3588 [RFC3588].

7. IANA Considerations

This document requires the assignment of two new RADIUS Attribute Types in the "Radius Types" registry (currently located at <http://www.iana.org/assignments/radius-types> for the following attributes:

- o 6rd

IANA should allocate these numbers from the standard RADIUS Attributes space using the "IETF Review" policy [RFC5226].

8. Acknowledgments

The authors would like to thank Maglione Roberta, Telecom Italia, for valuable comments.

9. Change Log [RFC Editor please remove]

draft-guo-softwire-6rd-radiusattrib-00, renaming and deleting DS-lite contents, 2010-10-18.

draft-guo-radext-softwire-concentrator-00, original version, 2010-07-05.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.
- [RFC3162] Aboba, B., Zorn, G., and D. Mitton, "RADIUS and IPv6", RFC 3162, August 2001.
- [RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J., Arkko, "Diameter Base Protocol", RFC 3588, September 2003.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC5226] T. Narten, H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, May 2008.
- [RFC5969] Townsley W., et al., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC5969, August 2010.

10.2. Informative References

- [RFC2607] Aboba, B. and J. Vollbrecht, "Proxy Chaining and Policy Implementation in Roaming", RFC 2607, June 1999.
- [RFC2869] Rigney, C., Willats, W., and P. Calhoun, "RADIUS Extensions", RFC 2869, June 2000.

Author's Addresses

Dayong Guo
Huawei Technologies Co., Ltd
Huawei Building, No.3 Xinxu Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing 100085
P.R. China
Email: guoseu@huawei.com

Sheng Jiang
Huawei Technologies Co., Ltd
Huawei Building, No.3 Xinxu Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing 100085
P.R. China
Email: shengjiang@huawei.com

Remi Despres
RD-IPtech
3 rue du President Wilson
Levallois,
France
Email: remi.despres@free.fr

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: January 14, 2011

Dayong Guo
Sheng Jiang
Huawei Technologies Co., Ltd
Brian Carpenter
University of Auckland
July 12, 2010

Software Concentrator Discovery Using DHCP

draft-guo-software-sc-discovery-04.txt

Status of this Memo

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

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

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

This Internet-Draft will expire on January 14, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Abstract

Several types of Carrier Grade NATs have been proposed to simplify IPv4/IPv6 transition of the edge network by integrating tunnels and NAT. A very common scenario is that many users set up softwires to a softwire concentrator for public or private access services. In order to establish softwires successfully, a new mechanism is required to enable users in the edge network to discover the information of the concentrator. This document describes how a host or Customer Premises Equipment discovers the remote softwire concentrator or CGN in a hub and spoke network using DHCP. Based on two new Softwire Concentrator Discovery DHCP Options, proposed in the document, a user can obtain the information of the softwire concentrator or CGN and then set up a tunnel to it.

Table of Contents

1. Introduction.....	3
2. Terminology.....	4
3. DHCP Solution Overview for Softwire Concentrator Discovery....	4
4. DHCPv4 Softwire Concentrator Discovery (SCD) Option.....	5
4.1. Suboptions in DHCPv4 SCD Option.....	6
4.1.1. Protocol Type Suboption.....	7
4.1.2. GRE Key Suboption.....	7
5. DHCPv6 Softwire Concentrator Discovery (SCD) Option.....	7
5.1. Suboptions in DHCPv6 SCD Option.....	8
5.1.1. Protocol Type Suboption.....	9
5.1.2. Prefix Suboption.....	10
5.1.3. GRE Key Suboption.....	10
6. Illustration Examples.....	11
6.1. Example 1: Incremental CGN scenario.....	11
6.2. Example 2: two CGN in DS lite scenario.....	11
7. Security Considerations.....	11
8. IANA Considerations.....	12
8.1. Tunnel Types.....	12
8.2. DHCPv4 SCD Suboption Types.....	12
8.3. DHCPv6 SCD Suboption Types.....	13
9. Acknowledgments.....	13
10. Change Log [RFC Editor please remove].....	13
11. References.....	13
11.1. Normative References.....	13
11.2. Informative References.....	14

1. Introduction

Transition is an important factor for user experience in IPv4 and IPv6 coexistence phase. The transition of the edge network is the most complicated because it is near lots of users and uses multiple network technologies. Recently, several types of Carrier-Grade-NATs (CGNs) have been proposed to simplify IPv4/IPv6 transition of the edge network by integrating tunnels and NAT. Incremental CGN [I-D.ietf-v6ops-incremental-cgn] and 6rd [I-D.ietf-software-ipv6-6rd] and describes how dispersed IPv6 users bridge with the IPv6 Internet by tunnel spanning ipv4 infrastructure. The dual-stack lite technology [I-D.ietf-software-dual-stack-lite] is intended for maintaining connectivity to legacy IPv4 devices and networks using IPv4-over-IPv6 softwires while a service provider deploys an IPv6-only network. A very common scenario is that many users set up softwires or tunnels to a software concentrator for public or private access services.

The aforementioned scenarios have been abstracted as hub and spoke networks in the IETF Software working group, and several encapsulation techniques have been defined [RFC4925] [RFC5512]. [RFC5571] discloses a mechanism in mesh network by BGP extension for users to discover the information of a tunnel end point. However, the nodes in an edge network do not have BGP capability generally. Manual configuration is not suitable because the address and other attribute of the concentrator may change. A new mechanism is required to enable users in edge network to discover the information of the concentrator automatically.

In order to establish a software successfully, users must know the information of a software concentrator or CGN, such as address, tunnel type. Additionally, the discovery process may also support multiple protocol type in tunnel, load-sharing and recovery from a single point of failure.

Since ISPs may use different software technologies, an ISP-independent CPE should support as many as possible potential software technologies and be able to auto discovery which software technologies is in use. Even within a single ISP, different software technologies may also use to differentiate customers, e.g., support of secured encapsulation for some customers and plain IP-in-IP encapsulation for others.

For scalability and stability purposes, customers may be assigned different/multiple software concentrators through the discovery mechanism.

The Dynamic Host Configuration Protocol (DHCP [RFC2131], [RFC3315]) is widely used in edge networks to enable auto-configuration. This document extends DHCP to support discovery of a softwire concentrator or CGN. This mechanism is general for 6rd, incremental CGN, DS-Lite and Port-range Router [I-D.boucadair-port-rang]. It can also be extended to support the discovery of other concentrators with tunnels.

In the absence of DHCP, PPP or Router Advertisements could be used to find a softwire concentrator or CGN automatically, but this document does not discuss these methods in detail.

2. Terminology

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

3. DHCP Solution Overview for Softwire Concentrator Discovery

In order to support softwire concentrator or CGN discovery, two new DHCP options are defined respectively for DHCPv4 and DHCPv6. They have the identical structure apart from address length.

When a DHCP server answers a client request message, softwire concentrator information can be carried in a DHCP reply message. Thus a client is configured the address and other attributes of a softwire concentrator or CGN and can automatically set up a tunnel.

DHCP server decides to attach SCD option based on policy. One choice is to respond only if the client requests the SCD option; another is to append it to every reply no matter the client requests the SCD option or not.

For load sharing or single-point failure recovery purposes, a DHCPv4 reply message may carry multiple instances in a single DHCPv4 SCD option; a DHCPv6 reply message may carry more than one DHCPv6 SCP options.

Section 4 defines a new DHCPv4 Softwire Concentrator Discovery (SCD) option while Section 5 defines DHCPv6 SCD option. Section 4.1 defines sub-options that apply to DHCPv4 SCD option while Section 5.1 defines sub-options that apply to DHCPv6 SCD option.

4. DHCPv4 Software Concentrator Discovery (SCD) Option

The DHCPv4 Software Concentrator Discovery (SCD) Option is mainly used when an IPv6 host or CPE in an IPv4 ISP network wants to obtain an IPv4 address of an IPv6 access point or an incremental CGN. The Option is carried in DHCPv4.

A DHCPv4 message can carry only one DHCPv4 SCD Option. Multiple instances can be concatenated in the DHCPv4 SCD Option, as follow:

Code	Len	Len	Data	Len	Data
TBD1	n	n1	data1...	n2	data2... ...

The DHCPv4 SCD Option 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	2	3	4	5	6	7	8	9
Code										Len										Instance1-Len										Tunnel Type									
Preference										Software Concentrator or CGN Address																													
cont.																																							
										Instance1's Suboptions																													
Instance2-Len																																							
										Instance2-Data																													
																				Instance n																			

Code TBD1.

Len n + Len1 + Len2 + ... + Len n.

Instance-Len 6 + length of Instance's sub options in octets.

Tunnel Type Tunnel type which users connect to software concentrators or CGN. A few initial value assignments, like L2TPv2, GRE, ISATAP, 6to4, 6rd, IPSec and other IP in IP, is listed in Section 8 IANA consideration.

Preference This indicates the preference level for a software concentrator or CGN. 0 is the highest. When receiving multiple instances, the user chooses a primary software concentrator among them based on the preference. The others are backup software concentrators. The service provider assigns different preference for each software concentrator to support traffic engineering.

Software Concentrator or CGN Address The outer layer IPv4 address of software concentrator, which is used to establish tunnel.

Sub Options An optional, variable length field which is defined in Section 4.1.

4.1. Suboptions in DHCPv4 SCD Option

The suboptions defined in this section can be applied to DHCPv4 SCD option, defined above. They are used to configure the complementary tunnel information.

The DHCPv4 SCD suboption is structured in TLV style 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
+-----+-----+-----+-----+-----+-----+-----+-----+
| Suboption Type| Suboption Len |                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
.                               Suboption Value (Variable)                               .
+-----+-----+-----+-----+-----+-----+-----+-----+

```

* DHCPv4 SCD Suboption Type (1 octet): each suboption type defines a certain property about the tunnel. The following are the types defined in this document:

- Protocol Type: suboption type = 0
- GRE Key: suboption type = 1

New suboptions may be defined in the future. Any unknown suboptions MUST be ignored and skipped.

* Suboption Length (1 octet): the total number of octets of the suboption value field.

- * Suboption Value (variable): encodings of the value field depend on the suboption type as enumerated above.

The following sub-sections define the encoding in detail.

4.1.1. Protocol Type Suboption

This suboption designates which protocol is encapsulated in tunnel.

```

      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 = 0   |   Len = 2   |           Protocol Type           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Protocol Type field is defined in [IANA-ET] as ETHER TYPEs. The most used protocols are IPv4 (0x0800) and IPv6 (0x86dd).

4.1.2. GRE Key Suboption

When the tunnel type is GRE, this suboption may be contained.

```

      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 = 1   |   Len = 4   |           GRE Key           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           GRE Key (cont.)           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

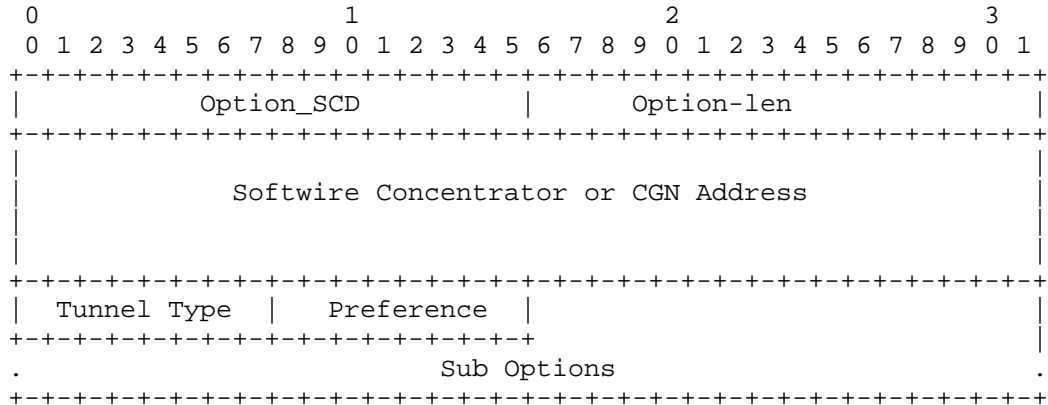
GRE Key: 4-octet field [RFC2890] that is generated by the Software Concentrator or CGN. If the client receives the GRE Key suboption, the key MUST be inserted into the GRE encapsulation header of the payload packets sent by the client to the Software Concentrator or CGN. It is used for identifying extra context information about the received payload. The payload packets without the correspondent GRE key or with an unmatched GRE Key will be silently dropped.

5. DHCPv6 Software Concentrator Discovery (SCD) Option

The DHCPv6 Software Concentrator Discovery (SCD) Option is mainly used when an IPv4 host or CPE in an IPv6 ISP network wants to learn an IPv6 address of an IPv4 access point or a DS-lite CGN. The Option is carried in DHCPv6.

A DHCPv6 Reply message can carry more than one SCD Options.

The DHCPv6 SCD Option is structured as follows:



Option-code Option_SCD (TBD2).

Option-len 18 + length of sub options in octets.

Software Concentrator or CGN Address The outer layer IPv6 address of software concentrator, which is used to establish tunnel.

Tunnel Type Tunnel type which users connect to software concentrators or CGN. A few initial value assignments, like L2TPv2, GRE, IPSec and IP in IP, is listed in Section 8 IANA consideration.

Preference This indicates the preference level for a software concentrator or CGN. 0 is the highest. When receiving multiple options, user chooses a primary software concentrator among them based on the preference. The others are backup software concentrators. The service provider assigns different preference of each software concentrator to support traffic engineering.

Sub Options An optional, variable length field is defined in Section 5.1.

5.1. Suboptions in DHCPv6 SCD Option

The suboptions defined in this section can be applied to DHCPv6 SCD option, defined above. They are used to configure the complementary tunnel information.

The DHCPv6 SCD suboption is structured in TLV style 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
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Suboption Type               |               Suboption Len               |
+-----+-----+-----+-----+-----+-----+-----+-----+
.               Suboption Value (Variable)               .
+-----+-----+-----+-----+-----+-----+-----+-----+

```

* DHCPv4 SCD Suboption Type (2 octet): each suboption type defines a certain property about the tunnel. The following are the types defined in this document:

- Protocol Type: suboption type = 0
- Prefix: suboption type = 1
- GRE Key: suboption type = 2

New suboptions may be defined in the future. Any unknown suboptions MUST be ignored and skipped.

* Suboption Length (2 octet): the total number of octets of the suboption value field.

* Suboption Value (variable): encodings of the value field depend on the suboption type as enumerated above.

The following sub-sections define the encoding in detail.

5.1.1.1. Protocol Type Suboption

This suboption designates which protocol is encapsulated in tunnel.

```

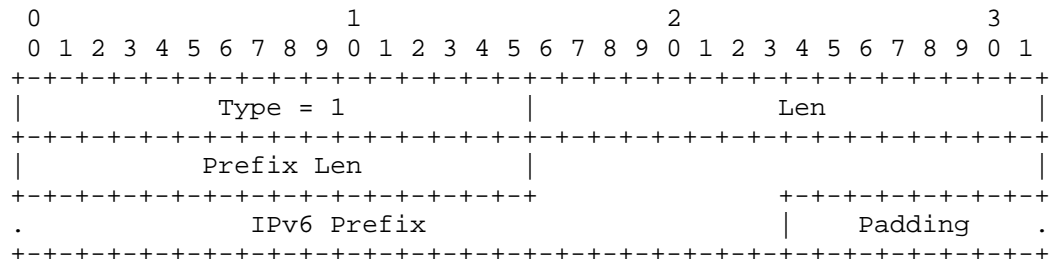
      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Type = 0               |               Len = 2               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Protocol Type               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Protocol Type field is defined in [IANA-ET] as ETHER TYPEs. The most used protocols are IPv4 (0x0800) and IPv6 (0x86dd).

5.1.2. Prefix Suboption

This suboption designates IPv6 prefix which is used to construct internal address of the tunnel.



Len: total length of the prefix and padding fields in octets.

Prefix Len: Length for this prefix in bits.

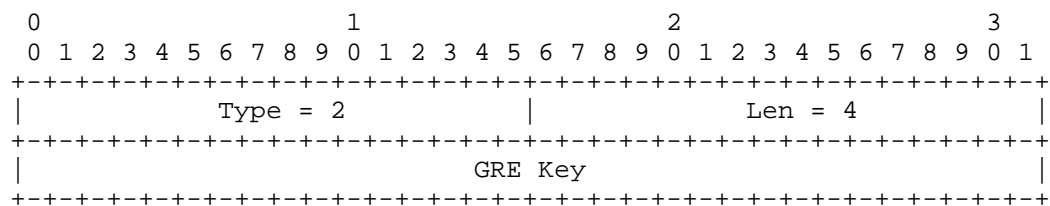
IPv6 Prefix: IPv6 prefix allocated to the client to construct internal address of the tunnel.

Padding: additional 0~7 bits MUST be padded at the end of IPv6 Prefix field when the value in Prefix Len field is not a multiple of 8-bit. The padding bits SHOULD be set as 0.

The semantics of the value field is determined by the tunnel type. For example, a client can obtain IPv6 Prefix of ISATAP tunnel by this suboption in DHCPv6 SDC Option.

5.1.3. GRE Key Suboption

When the tunnel type is GRE, this suboption may be contained.



GRE Key: 4-octet field [RFC2890] that is generated by the Software Concentrator or CGN. If the client receives the GRE Key suboption, the key **MUST** be inserted into the GRE encapsulation header of the payload packets sent by the client to the Software Concentrator or CGN. It is used for identifying extra context

information about the received payload. The payload packets without the correspondent GRE key or with an unmatched GRE Key will be silently dropped.

6. Illustration Examples

6.1. Example 1: Incremental CGN scenario

As an example, an incremental CGN with IP address 192.0.2.1 and L2TPv2 tunnel support is deployed in an IPv4 ISP network. The CGN information is stored in a DHCPv4 server. When a dual stack user in the network wants to connect IPv6 Internet, it will send a DHCPv4 request message to the DHCP server to obtain the CGN information. The DHCP server replies with a SCD option. The parameters in the SCD option are "CGN address = 192.0.2.1, tunnel type = 1, preference = 80". After the user receives the option, it can set up an L2TPv2 tunnel with the CGN.

6.2. Example 2: two CGN in DS lite scenario

In another example scenario, there are two DS lite CGNs deployed in order to provide redundancy and load balancing. DS lite CGN1 is 2001:db8:a::1, the other CGN2 is 2001:db8:b::1. Both of them support IPv4 in IPv6 tunnel. The preference of each CGN is decided by the network management policy. A user may get two SCD options, one describes CGN1 "CGN address = 2001:db8:a::1, tunnel type = 3, preference = 80" and the other describes CGN2 "CGN address = 2001:db8:b::1, tunnel type = 3, preference = 255". The user should establish an IPv4 in IPv6 tunnel with the CGN1, which has higher preference. When the CGN1 is down, the user may re-establish tunnel to the CGN2.

For the load balancing purpose, another user may receive the options, in which CGN2 has the higher preference value. The user may set CGN2 as its primary CGN.

7. Security Considerations

There are two forms of attack using bogus SCD options should be noticeable:

1. A wiretap attack, in which a bogus concentrator observes the traffic before pretending to be the real client and sending the traffic to the real concentrator.
2. A DoS attack, in which a bogus concentrator is used in some way to create a loop or simply to act as a source of DoS packets.

The mechanisms based on DHCPv6 are all vulnerable by man-in-middle attacks. Proper use of DHCPv6 auto-configuration facilities [RFC3315], such as AUTH option or Secure DHCPv6 [I-D.ietf-dhc-secure-dhcpv6] can prevent these threats, provided that a configuration token is known to both the client and the server.

8. IANA Considerations

IANA is requested to allocate one DHCPv4 SCD Option code TBD1 and one DHCPv6 Option code TBD2.

This document defines three new namespaces:

- Tunnel Types
- DHCPv4 SCD Suboption Types
- DHCPv6 SCD Suboption Types

8.1. Tunnel Types

Section 4 & 5 defines the following Tunnel Types, which should be assigned by IANA for use within DHCPv4 & DHCPv6 SCD Option. IANA set up a registry for "Tunnel Types for DHCP SCD Option". This is a registry of one-octet values (0-255), to be assigned on a first-come, first-served basis. The initial assignments are as follows:

Tunnel Name	Type
-----	-----
Reserved	0
L2TPv2	1
GRE	2
IP-in-IP	3
ISATAP	4
6to4	5
6rd	6
IPsec	7

8.2. DHCPv4 SCD Suboption Types

Section 4.1 defines the following SCD Suboption Types, which should be assigned by IANA for use within DHCPv4 SCD Option. IANA set up a registry for "DHCPv4 SCD Suboption Types". This is a registry of one-octet values (0-255), to be assigned on a first-come, first-served basis. The initial assignments are as follows:

Tunnel Name	Type
-----	-----
Protocol Type	0
GRE Key	1

8.3. DHCPv6 SCD Suboption Types

Section 5.1 defines the following SCD Suboption Types, which should be assigned by IANA for use within DHCPv6 SCD Option. IANA set up a registry for "DHCPv6 SCD Suboption Types". This is a registry of one-octet values (0-255), to be assigned on a first-come, first-served basis. The initial assignments are as follows:

Tunnel Name	Type
-----	-----
Protocol Type	0
Prefix	1
GRE Key	2

9. Acknowledgments

The authors would like to thank Wei Cao, Huawei, Bernie Volz, Cisco for valuable comments.

10. Change Log [RFC Editor please remove]

draft-guo-software-sc-discovery-00, original version, 2009-06-23.

draft-guo-software-sc-discovery-01, revised for protocol type, 2009-07-13.

draft-guo-software-sc-discovery-02, revised after comments at IETF75 and comments on the maillist, 2009-10-26.

draft-guo-software-sc-discovery-03, minor update, 2010-03-05.

draft-guo-software-sc-discovery-04, revised after comments at IETF77 and comments on the maillist, 2010-07-12.

11. References

11.1. Normative References

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

- [RFC2131] R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC2890] G. Dommety, "Key and Sequence Number Extensions to GRE", RFC 2890, September 2000.
- [RFC3315] R. Droms, et al., "Dynamic Host Configure Protocol for IPv6", RFC3315, July 2003.
- [RFC5512] P. Mohapatra, E. and Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, April 2009.
- [RFC5571] B. Storer, et al., "Softwire Hub & Spoke Deployment Framework with L2TPv2", RFC 5571, June 2009.

11.2. Informative References

- [RFC4925] X. Li, S. Dawkins, D. Ward, and A. Durand, "Softwire Problem Statement", RFC 4925, July 2007.
- [I-D.ietf-softwire-dual-stack-lite]
A. Durand, R. Droms, B. Haberman, and J. Woodyatt, "Dual-stack lite broadband deployments post IPv4 exhaustion", draft-ietf-softwire-dual-stack-lite, work in progress, March 2010.
- [I-D.ietf-v6ops-incremental-cgn]
S. Jiang, D. Guo, and B. Carpenter, "An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition" draft-ietf-v6ops-incremental-cgn, work in progress, June 2010.
- [I-D.ietf-dhc-secure-dhcpv6]
S. Jiang and S. Shen, "Secure DHCPv6 Using CGAs", draft-ietf-dhc-secure-dhcpv6, work in progress, June 2010.
- [I-D.ietf-softwire-ipv6-6rd]
Townesley W., et al., "IPv6 via IPv4 Service Provider Networks (6rd)", draft-ietf-softwire-ipv6-6rd, (work in progress), March 2010.
- [I-D.boucadair-port-rang]
B. Storer, et al., "IPv4 Connectivity Access in the Context of IPv4 Address Exhaustion", draft-boucadair-port-range-02.txt, work in progress, July 2009.

[IANA-ET] "Ether Types", <http://www.iana.org/assignments/ethernet-numbers>.

Author's Addresses

Dayong Guo
Huawei Technologies Co., Ltd
Huawei Building, No.3 Xinxu Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing 100085
P.R. China
Email: guoseu@huawei.com

Sheng Jiang
Huawei Technologies Co., Ltd
Huawei Building, No.3 Xinxu Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing 100085
P.R. China
Email: shengjiang@huawei.com

Brian Carpenter
Department of Computer Science
University of Auckland
PB 92019
Auckland, 1142
New Zealand
Email: brian.e.carpenter@gmail.com

software
Internet-Draft
Intended status: Standards Track
Expires: April 15, 2011

R. Maglione
Telecom Italia
A. Durand
Juniper Networks
October 12, 2010

RADIUS Extensions for Dual-Stack Lite
draft-ietf-software-dslite-radius-ext-00

Abstract

Dual-Stack Lite is a solution to offer both IPv4 and IPv6 connectivity to customers which are addressed only with an IPv6 prefix. DS-Lite requires to pre-configure the AFTR tunnel information on the B4 element. In many networks, the customer profile information may be stored in AAA servers while client configurations are mainly provided through DHC protocol. This document specifies two new RADIUS attributes to carry Dual-Stack Lite Address Family Transition Router (AFTR) IPv6 address and name; the RADIUS attributes are defined based on the equivalent DHCPv6 options already specified in draft-ietf-software-ds-lite-tunnel-option. These RADIUS attributes are meant to be used between the RADIUS Server and the NAS, they are not intended to be used directly between the B4 element and the RADIUS Server.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 15, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

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

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
2. Terminology	5
3. DS-Lite Configuration with RADIUS and DHCPv6	5
4. Attributes	6
4.1. DS-Lite-Tunnel-Addr	6
4.2. DS-Lite-Tunnel-Name	6
5. Table of attributes	7
6. Security Considerations	8
7. IANA Considerations	8
8. Normative References	8
Authors' Addresses	9

1. Introduction

Dual-Stack Lite [I-D.ietf-softwire-dual-stack-lite] is a solution to offer both IPv4 and IPv6 connectivity to customers which are addressed only with an IPv6 prefix (no IPv4 address is assigned to the attachment device). One of its key components is an IPv4-over-IPv6 tunnel, but a DS-Lite Basic Bridging BroadBand (B4) will not know if the network it is attached to offers Dual-Stack Lite support, and if it did, would not know the remote end of the tunnel to establish a connection.

To inform the B4 of the AFTR's location, either an IPv6 address or Fully Qualified Domain Name (FQDN) may be used. Once this information is conveyed, the presence of the configuration indicating the AFTR's location also informs a host to initiate Dual-Stack Lite (DS-Lite) service and become a Softwire Initiator.

The draft draft-ietf-softwire-ds-lite-tunnel-option [I-D.ietf-softwire-ds-lite-tunnel-option] specifies two DHCPv6 options which are meant to be used by a Dual-Stack Lite client (Basic Bridging BroadBand element, B4) to discover its Address Family Transition Router (AFTR) address. In order to be able to populate such options the DHCPv6 Server must be pre-provisioned with the Address Family Transition Router (AFTR) address or name.

In Broadband environments, customer profile may be managed by AAA servers, together with user Authentication, Authorization, and Accounting (AAA). RADIUS protocol [RFC2865] is usually used by AAA Servers to communicate with network elements. [I-D.ietf-radext-ipv6-access] describes a typical broadband network scenario in which the Network Access Server (NAS) acts as the access gateway for the users (hosts or CPEs) and the NAS embeds a DHCPv6 Server function that allows it to locally handle any DHCPv6 requests issued by the clients.

Since the DS-Lite AFTR information can be stored in AAA servers and the client configuration is mainly provided through DHC protocol running between the NAS and the requesting clients, new RADIUS attributes are needed to send AFTR information from AAA server to the NAS.

This document aims at defining two new RADIUS attributes to be used for carrying the DS-Lite Tunnel Name and DS-Lite Tunnel Address, based on the equivalent DHCPv6 options already specified in [I-D.ietf-softwire-ds-lite-tunnel-option]

2. Terminology

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

The terms DS-Lite Basic Bridging BroadBand element (B4) and the DS-Lite Address Family Transition Router element (AFTR) are defined in [I-D.ietf-softwire-dual-stack-lite]

3. DS-Lite Configuration with RADIUS and DHCPv6

The Figure 1 illustrates how the RADIUS protocol and DHCPv6 work together to accomplish DS-Lite configuration on the B4 element.

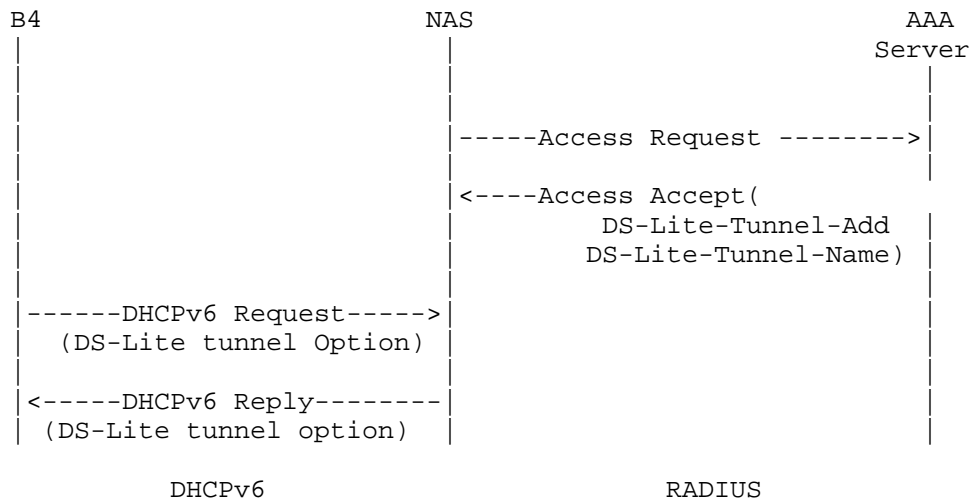


Figure 1: RADIUS and DHCPv6 Message Flow

The Network Access Server (NAS) operates as a client of RADIUS and as DHCP Server for DHC protocol. The NAS initially 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 also contain the AFTR Tunnel IPv6 Address and/or the AFTR Tunnel Name. When the NAS receives a DHCPv6 client request containing the DS-Lite tunnel Option, the NAS shall use the address returned in the RADIUS DS-Lite-Tunnel-Addr attribute to populate the DHCPv6 OPTION_DS_LITE_ADDR option in the DHCPv6 reply

message.

4. Attributes

This section specifies the format of the two new RADIUS attributes.

4.1. DS-Lite-Tunnel-Addr

Description

The DS-Lite-Tunnel-Addr RADIUS attribute contains a 128 bit IPv6 address that identifies the location of the remote tunnel endpoint, expected to be located at an AFTR. The NAS shall use the address returned in the RADIUS DS-Lite-Tunnel-Addr attribute to populate the DHCPv6 OPTION_DS_LITE_ADDR option [I-D.ietf-software-ds-lite-tunnel-option].

This attribute MAY be used in Access-Accept packets and it MAY be present in Accounting-Request records where the Acct-Status-Type is set to Start, Stop or Interim-Update. The DS-Lite-Tunnel-Addr RADIUS attribute and MUST NOT appear more than once in a message.

A summary of the DS-Lite-Tunnel-Addr RADIUS attribute format is shown below. The fields are transmitted from left to right.

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	2	3	4	5	6	7	8	9																								
Type																Length																DS-Lite-Tunnel-Addr																															
DS-Lite-Tunnel-Addr (IPv6 Address)(cont)																																																															

Type:

TBA1 for DS-Lite-Tunnel-Addr.

Length:

16 octets

DS-Lite-Tunnel-Addr:

A 128-bit IPv6 address of the DS-Lite AFTR.

4.2. DS-Lite-Tunnel-Name

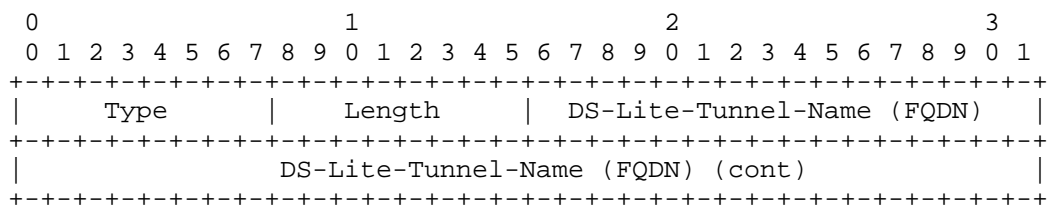
Description

The DS-Lite-Tunnel-Name RADIUS attribute contains a Fully Qualified

Domain Name that refers to the AFTR the client is requested to establish a connection with. The NAS shall use the name returned in the RADIUS DS-Lite-Tunnel-Name attribute to populate the DHCPv6 OPTION_DS_LITE_NAME option [I-D.ietf-softwire-ds-lite-tunnel-option]

This attribute MAY be used in Access-Accept packets and it MAY be present in Accounting-Request records where the Acct-Status-Type is set to Start, Stop or Interim-Update. The DS-Lite-Tunnel-Name RADIUS attribute and MUST NOT appear more than once in a message.

A summary of the DS-Lite-Tunnel-Name RADIUS attribute format is shown below. The fields are transmitted from left to right.



Type:

TBA2 for DS-Lite-Tunnel-Name.

Length:

Length in octets of the DS-Lite-Tunnel-Name (FQDN)

DS-Lite-Tunnel-Name:

A single Fully Qualified Domain Name of the remote tunnel endpoint, located at the DS-Lite AFTR.

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	#	Attribute
				Request		
0-1	0-1	0	0	0-1	TBA1	DS-Lite-Tunnel-Addr
0-1	0-1	0	0	0-1	TBA2	DS-Lite-Tunnel-Name

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.

6. Security Considerations

This document has no additional security considerations beyond those already identified in [RFC2865]

[I-D.ietf-softwire-dual-stack-lite] discusses DS-Lite related security issues.

7. IANA Considerations

This document requests the allocation of two new Radius attribute types from the IANA registry "Radius Attribute Types" located at <http://www.iana.org/assignments/radius-types>

DS-Lite-Tunnel-Addr - TBA1
DS-Lite-Tunnel-Name - TBA2

8. Normative References

- [I-D.ietf-radext-ipv6-access]
Lourdelet, B., Dec, W., Sarikaya, B., Zorn, G., and D. Miles, "RADIUS attributes for IPv6 Access Networks", draft-ietf-radext-ipv6-access-02 (work in progress), July 2010.
- [I-D.ietf-softwire-ds-lite-tunnel-option]
Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual- Stack Lite", draft-ietf-softwire-ds-lite-tunnel-option-05 (work in progress), September 2010.
- [I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-06 (work in progress), August 2010.
- [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.

Authors' Addresses

Roberta Maglione
Telecom Italia
Via Reiss Romoli 274
Torino 10148
Italy

Phone:
Email: roberta.maglione@telecomitalia.it

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

Phone:
Fax:
Email: adurand@juniper.net
URI:

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 28, 2011

F. Brockners
S. Gundavelli
Cisco
S. Speicher
Deutsche Telekom AG
D. Ward
Juniper Networks
October 25, 2010

Gateway Initiated Dual-Stack Lite Deployment
draft-ietf-softwire-gateway-init-ds-lite-02

Abstract

Gateway-Initiated Dual-Stack lite (GI-DS-lite) is a variant of Dual-Stack lite (DS-lite) applicable to certain tunnel-based access architectures. GI-DS-lite extends existing access tunnels beyond the access gateway to an IPv4-IPv4 NAT using softwires with an embedded context identifier that uniquely identifies the end-system the tunneled packets belong to. The access gateway determines which portion of the traffic requires NAT using local policies and sends/receives this portion to/from this softwire.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 28, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Overview	3
2. Conventions	3
3. Gateway Initiated DS-Lite	4
4. Protocol and related Considerations	6
5. Software Management and related Considerations	7
6. Software Embodiments	7
7. GI-DS-lite deployment	9
7.1. Connectivity establishment: Example call flow	9
7.2. GI-DS-lite applicability: Examples	10
8. Acknowledgements	11
9. IANA Considerations	11
10. Security Considerations	11
11. Change History (to be removed prior to publication as an RFC)	11
12. References	12
12.1. Normative References	12
12.2. Informative References	13
Authors' Addresses	14

1. Overview

Gateway-Initiated Dual-Stack lite (GI-DS-lite) is a variant of the Dual-Stack lite (DS-lite) [I-D.ietf-software-dual-stack-lite], applicable to network architectures which use point to point tunnels between the access device and the access gateway. The access gateway in these models is designed to serve large numbers of access devices. Mobile architectures based on Mobile IPv6 [RFC3775], Proxy Mobile IPv6 [RFC5213], or GTP [TS29060], as well as broadband architectures based on PPP or point-to-point VLANs as defined by the Broadband Forum (see [TR59] and [TR101]) are examples for this type of architecture.

The DS-lite approach leverages IPv4-in-IPv6 tunnels (or other tunneling modes) for carrying the IPv4 traffic from the customer network to the Address Family Transition Router (AFTR). An established software between the AFTR and the access device is used for traffic forwarding purposes. This turns the inner IPv4 address irrelevant for traffic routing and allows sharing private IPv4 addresses [RFC1918] between customer sites within the service provider network.

Similar to DS-lite, GI-DS-lite enables the service provider to share public IPv4 addresses among different customers by combining tunneling and NAT. It allows multiple access devices behind the access gateway to share the same private IPv4 address [RFC1918]. Rather than initiating the tunnel right on the access device, GI-DS-lite logically extends the already existing access tunnels beyond the access gateway towards the IPv4-IPv4 NAT using a tunneling mechanism with semantics for carrying context state related to the encapsulated traffic. This approach results in supporting overlapping IPv4 addresses in the access network, requiring no changes to either the access device, or to the access architecture. Additional tunneling overhead in the access network is also omitted. If e.g., a GRE based encapsulation mechanism is chosen, it allows the network between the access gateway and the NAT to be either IPv4 or IPv6 and provides the operator to migrate to IPv6 in incremental steps.

2. Conventions

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

The following abbreviations are used within this document:

AFTR: Address Family Transition Router (also known as "Large Scale NAT (LSN)" or "Dual-Stack lite Tunnel Concentrator", or "Carrier Grade NAT"). An AFTR combines IP-in-IP tunnel termination and IPv4-IPv4 NAT.

AD: Access Device. It is the end host, also known as the mobile node in mobile architectures.

CID: Context Identifier

DS-lite: Dual-stack lite

GI-DS-lite: Gateway-initiated DS-lite

NAT: Network Address Translator

SW: Softwire (see [RFC4925])

SWID: Softwire Identifier

TID: Access Tunnel Identifier. The interface identifier of the point-to-point access tunnel.

3. Gateway Initiated DS-Lite

The section provides an overview of Gateway Initiated DS-Lite (GI-DS-lite). Figure 1 outlines the generic deployment scenario for GI-DS-lite. This generic scenario can be mapped to multiple different access architectures, some of which are described in Section 7.

In Figure 1, access devices (AD-1 and AD-2) are connected to the Gateway using some form of tunnel technology and the same is used for carrying IPv4 (and optionally IPv6) traffic of the access device. These access devices may also be connected to the Gateway over point-to-point links. The details on how the network delivers the IPv4 address configuration to the access devices are specific to the access architecture and are outside the scope of this document. With GI-DS-lite, Gateway and AFTR are connected by a softwire [RFC4925]. The softwire is identified by a softwire identifier (SWID). The form of the SWID depends on the tunneling technology used for the softwire. The SWID could e.g. be the endpoints of a GRE-tunnel or a VPN-ID, see Section 6 for details. A Context-Identifier (CID) is used to multiplex flows associated with the individual access devices onto the softwire. Local policies at the Gateway determine which part of the traffic received from an access device is tunneled over the softwire to the AFTR. The combination of CID and SWID (potentially along with other traffic identifiers such as e.g.

interface, VLAN, port, etc.) serves as common context between Gateway and AFTR to uniquely identify flows associated with an access device. The CID is typically a 32-bit wide identifier and is assigned by the Gateway. It is retrieved either from a local or remote (e.g. AAA) repository. Like the SWID, the embodiment of the CID depends on the tunnel mode used and the type of the network connecting Gateway and AFTR. If, for example GRE [RFC2784] with "GRE Key and Sequence Number Extensions" [RFC2890] is used as software technology, the network connecting Gateway and AFTR could be either IPv4-only, IPv6-only, or a dual-stack IP network. The CID would be carried within the GRE-key field. See Section 6 for details on different software types supported with GI-DS-lite.

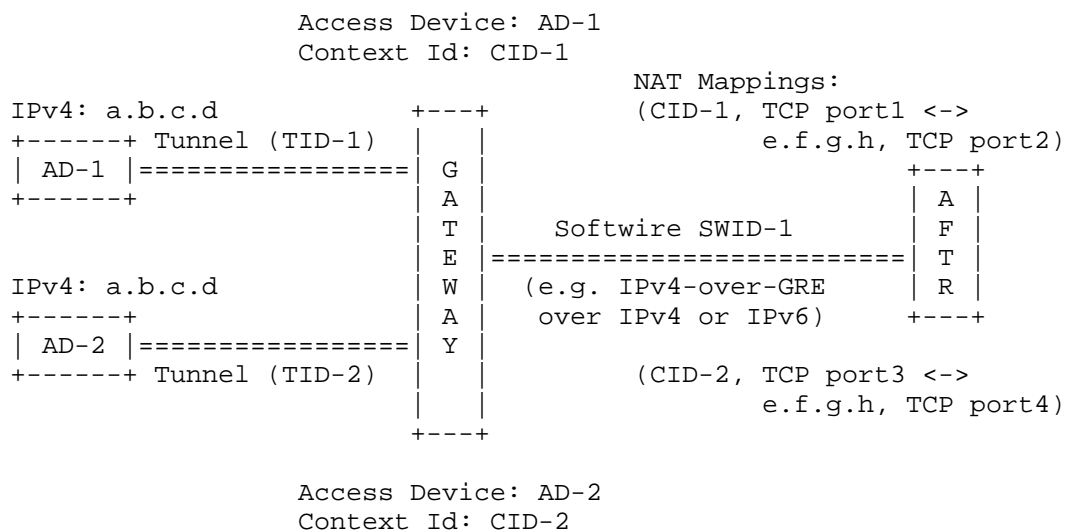


Figure 1: Gateway-initiated dual-stack lite reference architecture

The AFTR combines software termination and IPv4-IPv4 NAT. The outer/external IPv4 address of a NAT-binding at the AFTR is either assigned autonomously by the AFTR from a local address pool, configured on a per-binding basis (either by a remote control entity through a NAT control protocol or through manual configuration), or derived from the CID (e.g., the CID, in case 32-bit wide, could be mapped 1:1 to an external IPv4-address). A simple example of a translation table at the AFTR is shown in Figure 2. The choice of the appropriate translation scheme for a traffic flow can take parameters such as destination IP-address, incoming interface, etc. into account. The IP-address of the AFTR, which, depending on the transport network between the Gateway and the AFTR, will either be an IPv6 or an IPv4 address, is configured on the Gateway. A variety of methods, such as

out-of-band mechanisms, or manual configuration apply.

Software-Id/Context-Id/IPv4/Port	Public IPv4/Port
SWID-1/CID-1/a.b.c.d/TCP-port1	e.f.g.h/TCP-port2
SWID-1/CID-2/a.b.c.d/TCP-port3	e.f.g.h/TCP-port4

Figure 2: Example translation table on the AFTR

GI-DS-lite does not require a 1:1 relationship between Gateway and AFTR, but more generally applies to (M:N) scenarios, where M Gateways are connected to N AFTRs. Multiple Gateways could be served by a single AFTR. AFTRs could be dedicated to specific groups of access-devices, groups of Gateways, or geographic regions. An AFTR could, but does not have to be co-located with a Gateway.

4. Protocol and related Considerations

- o The NAT binding entry maintained at the AFTR, which reflects an active flow between an access device inside the network and a node in the Internet, needs to be extended to include two other parameters, the CID and the identifier of the software (SWID).
- o When creating an IPv4 to IPv4 NAT binding for an IPv4 packet flow received from the Gateway over the software, the AFTR will associate the CID with that NAT binding. It will use the combination of CID and SWID as the unique identifier and will store it in the NAT binding entry.
- o When forwarding a packet to the access device, the AFTR will obtain the CID from the NAT binding associated with that flow. E.g., in case of GRE-encapsulation, it will add the CID to the GRE Key and Sequence number extension of the GRE header and tunnel it to the Gateway.
- o On receiving any packet from the software, the AFTR will obtain the CID from the incoming packet and will use it for performing the NAT binding look up and for performing the packet translation before forwarding the packet.
- o The Gateway, on receiving any IPv4 packet from the access device will lookup the CID for that access device. In case of GRE

encapsulation it will for example add the CID to the GRE Key and Sequence number extension of the GRE header and tunnel it to the AFTR.

- o On receiving any packet from the softwire, the Gateway will obtain the CID from the packet and will use it for making the forwarding decision. There will be an association between the CID and the forwarding state.
- o When encapsulating an IPv4 packet, Gateway and AFTR can use its Diffserv Codepoint (DSCP) to derive the DSCP (or MPLS Traffic-Class Field in case of MPLS) of the softwire.

5. Softwire Management and related Considerations

The following are the considerations related to the operational management of the softwire between AFTR and Gateway.

- o The softwire between the Gateway and the AFTR is created at system startup time and stays up active all time. Deployment dependent, Gateway and AFTR can employ OAM mechanisms such as ICMP, BFD [RFC5880], or LSP ping [RFC4379] for softwire health management and corresponding protection strategies.
- o The softwire peers may be provisioned to perform policy enforcement, such as for determining the protocol-type or overall portion of traffic that gets tunneled, or for any other quality of service related settings. The specific details on how this is achieved or the types of policies that can be applied are outside the scope for this document.
- o The softwire peers must have a proper understanding of the path MTU value. This can be statically configured at softwire creation time.
- o A Gateway and an AFTR can have multiple softwires established between them (e.g. to separate address domains, provide for load-sharing etc.).

6. Softwire Embodiments

Deployment and requirements dependent, different tunnel technologies apply for the softwire connecting Gateway and AFTR. GRE encapsulation with GRE-key extensions, MPLS VPNs, or plain IP-in-IP encapsulation can be used. Softwire identification and Context-ID depend on the tunneling technology employed:

- o GRE with GRE-key extensions: Software identification is supplied by the endpoints of the GRE tunnel. The GRE-key serves as CID.
- o MPLS VPN: Software identification is supplied by the VPN identifier of the MPLS VPN. The IPv4-address serves as CID. The IPv4-address within a VPN has to be unique.
- o Plain IP-in-IP: Software identification is supplied by the endpoints of the IP-in-IP tunnel. Either the inner IPv4-address serves as CID (in which case the IPv4-address has to be unique) or the IPv6-Flow-Label serves as CID (which obviously only applies to cases where IPv6 transport is used). Note that when using the IP-Flow-Label as CID additional scaling considerations might apply given that the CID is to only 20 bits wide in this case. Also one should ensure sufficient randomization in this case to for example avoid interference with other uses of the IP-Flow-Label, such as ECMP.

Figure 3 gives an overview of the different tunnel modes as they apply to different deployment scenarios. "x" indicates that a certain deployment scenario is supported. The following abbreviations are used:

- o IPv4 address
 - * "up": Deployments with "unique private IPv4 addresses" assigned to the access devices are supported.
 - * "op": Deployments with "overlapping private IPv4 addresses" assigned to the access devices are supported.
 - * "nm": Deployments with "non-meaningful/dummy but unique IPv4 addresses" assigned to the access devices are supported.
 - * "s": Deployments where all access devices are assigned the same IPv4 address are supported.
- o Network-type
 - * "v4": Gateway and AFTR are connected by an IPv4-only network
 - * "v6": Gateway and AFTR are connected by an IPv6-only network
 - * "v4v6": Gateway and AFTR are connected by a dual stack network, supporting IPv4 and IPv6.
 - * "MPLS": Gateway and AFTR are connected by a MPLS network

Software	IPv4 address				Network-type			
	up	op	nm	s	v4	v6	v4v6	MPLS
GRE with GRE-key	x	x	x	x	x	x	x	
MPLS VPN	x	x	x					x
Plain IP-in-IP	x	x	x	x	x	x	x	

Figure 3: Tunnel modes and their applicability

Note: For "Plain IP-in-IP", support for 'op' and 's' requires the use of IPv6-transport with the IPv6-Flow-Label serving as CID.

7. GI-DS-lite deployment

7.1. Connectivity establishment: Example call flow

Figure 4 shows an example call flow - linking access tunnel establishment on the Gateway with the software to the AFTR. This simple example assumes that traffic from the AD uses a single access tunnel and that the Gateway will use local policies to decide which portion of the traffic received over this access tunnel needs to be forwarded to the AFTR.

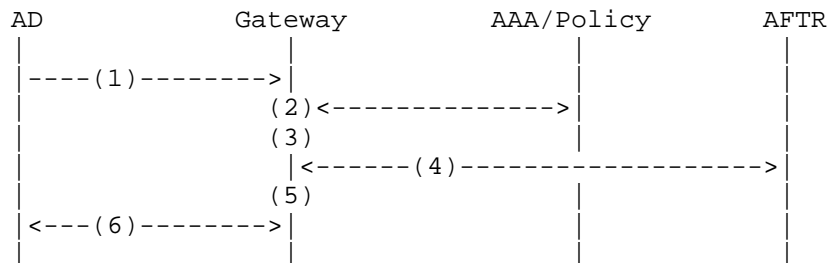


Figure 4: Example call flow for session establishment

1. Gateway receives a request to create an access tunnel endpoint.
2. The Gateway authenticates and authorizes the access tunnel. Based on local policy or through interaction with the AAA/Policy system the Gateway recognizes that IPv4 service should be provided using GI-DS-lite.

3. The Gateway creates an access tunnel endpoint. The access tunnel links AD and Gateway and is uniquely identified by Tunnel Identifier (TID) on the Gateway.
4. (Optional): The Gateway and the AFTR establish a control session between each other. This session can for example be used to exchange accounting or NAT-configuration information. Accounting information could be supplied to the Gateway, AAA/Policy, or other network entities which require information about the externally visible address/port pairs of a particular access device. The Diameter NAT Control Application (see [I-D.draft-ietf-dime-nat-control]) could for example be used for this purpose.
5. The Gateway allocates a unique CID and associates those flows received from the access tunnel (identified by the TID) that need to be tunneled towards the AFTR with the software linking Gateway and AFTR. Local forwarding policy on the Gateway determines which traffic will need to be tunneled towards the AFTR.
6. Gateway and AD complete the access tunnel establishment (depending on the procedures and mechanisms of the corresponding access network architecture this step can include the assignment of an IPv4 address to the AD).

7.2. GI-DS-lite applicability: Examples

The section outlines deployment examples of the generic GI-DS-lite architecture described in Section 3.

- o Mobile IP based access architectures: In a MIPv6 [RFC5555] based network scenario, the Mobile IPv6 home agent will implement the GI-DS-lite Gateway function along with the dual-stack Mobile IPv6 functionality.
- o Proxy Mobile IP based access architectures: In a PMIPv6 [RFC5213] scenario the local mobility anchor (LMA) will implement the GI-DS-lite Gateway function along with the PMIPv6 IPv4 support functionality.
- o GTP based access architectures: 3GPP TS 23.401 [TS23401] and 3GPP TS 23.060 [TS23060] define mobile access architectures using GTP. For GI-DS-lite, the PDN-Gateway/GGSN will also assume the Gateway function.
- o Fixed WiMAX architecture: If GI-DS-lite is applied to fixed WiMAX, the ASN-Gateway will implement the GI-DS-lite Gateway function.

- o Mobile WiMAX: If GI-DS-lite is applied to mobile WiMAX, the home agent will implement the Gateway function.
- o PPP-based broadband access architectures: If GI-DS-lite is applied to PPP-based access architectures the Broadband Remote Access Server (BRAS) or Broadband Network Gateway (BNG) will implement the GI-DS-lite Gateway function.
- o In broadband access architectures using per-subscriber VLANs the BNG will implement the GI-DS-lite Gateway function.

8. Acknowledgements

The authors would like to acknowledge the discussions on this topic with Mark Grayson, Jay Iyer, Kent Leung, Vojislav Vucetic, Flemming Andreassen, Dan Wing, Jouni Korhonen, Teemu Savolainen, Parviz Yegani, Farooq Bari, Mohamed Boucadair, Vinod Pandey, Jari Arkko, Eric Voit, Yiu L. Lee, Tina Tsou, Guo-Liang Yang, and Cathy Zhou.

9. IANA Considerations

This document includes no request to IANA.

All drafts are required to have an IANA considerations section (see the update of RFC 2434 [RFC5226] for a guide). If the draft does not require IANA to do anything, the section contains an explicit statement that this is the case (as above). If there are no requirements for IANA, the section will be removed during conversion into an RFC by the RFC Editor.

10. Security Considerations

All the security considerations from GTP [TS29060], Mobile IPv6 [RFC3775], Proxy Mobile IPv6 [RFC5213], and Dual-Stack lite [I-D.ietf-softwire-dual-stack-lite] apply to this specification as well.

11. Change History (to be removed prior to publication as an RFC)

Changes from -00 to -01

- a. clarified the applicability of GI-DS-lite to scenarios with M Gateways and N AFTRs.

- b. clarification of the nomenclature and use of the identifier of the software connecting Gateway and AFTR: Introduced software identifier (SWID), updated figure 2 accordingly.
- c. cleanup of editorial nits.
- d. added IP-Flow-Label as CID.

Changes from -00 to -02

- a. added considerations for the use of the IP-Flow-Label as CID.
- b. editorial edits (additional acknowledgements).

12. References

12.1. Normative References

- [I-D.ietf-software-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-software-dual-stack-lite-06 (work in progress), August 2010.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.
- [RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, September 2000.
- [RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.
- [RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.
- [RFC5213] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.

- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5555] Soliman, H., "Mobile IPv6 Support for Dual Stack Hosts and Routers", RFC 5555, June 2009.
- [RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, June 2009.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

12.2. Informative References

- [I-D.draft-ietf-dime-nat-control]
Brockners, F., Bhandari, S., Singh, V., and V. Fajardo,
"Diameter NAT Control Application", August 2009.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.
- [RFC4925] Li, X., Dawkins, S., Ward, D., and A. Durand, "Softwire Problem Statement", RFC 4925, July 2007.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.
- [TR101] Broadband Forum, "TR-101: Migration to Ethernet-Based DSL Aggregation", April 2006.
- [TR59] Broadband Forum, "TR-059: DSL Evolution - Architecture Requirements for the Support of QoS-Enabled IP Services", September 2003.
- [TS23060] "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS); Service description; Stage 2.", 2009.
- [TS23401] "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN)

access.", 2009.

[TS29060] "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP), V9.1.0", 2009.

Authors' Addresses

Frank Brockners
Cisco
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Sri Gundavelli
Cisco
170 West Tasman Drive
SAN JOSE, CA 95134
USA

Email: sgundave@cisco.com

Sebastian Speicher
Deutsche Telekom AG
Landgrabenweg 151
BONN, NORDRHEIN-WESTFALEN 53277
Germany

Email: sebastian.speicher@telekom.de

David Ward
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, California 94089-1206
USA

Email: dward@juniper.net

Softwire
Internet-Draft
Intended status: Informational
Expires: April 18, 2011

Y. Lee
Comcast
R. Maglione
Telecom Italia
C. Williams
MCSR Labs
C. Jacquenet
M. Boucadair
France Telecom
October 15, 2010

Deployment Considerations for Dual-Stack Lite
draft-lee-softwire-dslite-deployment-00

Abstract

This document discusses the deployment issues and describes requirements for the deployment and operation of Dual-Stack Lite. This document describes the various deployment scenarios and applicability of the Dual-Stack Lite protocol.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 18, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Overview	3
2. AFTR Deployment Considerations	3
2.1. MTU Considerations	3
2.2. Lawful Intercept Considerations	4
2.3. Logging at the AFTR	4
2.3.1. AFTR's Policies	5
2.4. AFTR Impacts on Internal Accounting Systems	5
2.4.1. AFTR Impacts on Accounting Process in Broadband Access	5
2.5. Reliability Considerations of AFTR	6
2.6. Strategic Placement of AFTR	6
2.7. AFTR Considerations for Geographically Aware Services . .	7
2.8. Impacts on QoS	7
2.9. Port Forwarding Considerations	7
3. B4 Deployment Considerations	7
3.1. DNS deployment Considerations	8
4. Security Considerations	8
5. Conclusion	9
6. Acknowledgement	9
7. IANA Considerations	9
8. References	9
8.1. Normative References	9
8.2. Informative References	10
Authors' Addresses	10

1. Overview

Dual-stack Lite (DS-Lite) [I-D.ietf-softwire-dual-stack-lite] is a transition technique that enable operators to multiplex public IPv4 addresses while provisioning only IPv6 to users. DS-Lite is designed to address the IPv4 depletion issue and allow the operators to upgrade their network incrementally to IPv6. DS-Lite combines IPv4-in-IPv6 tunnel and NAT44 to share a public IPv4 address more than one user. This document discusses various deployment considerations for DS-Lite by operators.

2. AFTR Deployment Considerations

Address Family Transition Router (AFTR) is the function deployed inside the operator's network. AFTR can be a standalone device or embedded into a router. AFR is the IPv4-in-IPv6 tunnel termination point and the NAT44 device. It is deployed at the IPv4-IPv6 network border where the tunnel interface is IPv6 and the NAT interface is IPv4. Although an operator can configure a dual-stack interface for both functions, we strongly recommended to configure two individual interfaces (i.e. one dedicated for IPv4 and one dedicated for IPv6) to segregate the functions.

In this section, the deployment considerations for AFTR are described.

2.1. MTU Considerations

DS-Lite is part tunneling protocol. Tunneling introduces some additional complexity and has a risk of MTU or other mis-configurations. With tunneling comes additional header overhead that implies that the tunnel's MTU is smaller than the raw interface MTU. The second problem is that between the B4 and AFTR networking entities there may exist further tunnels inside tunnel, so that the tunnel ingress is not necessarily aware of the true tunnel MTU. The third problem is that the routing of the interior of the tunnel may change, so that the tunnel MTU may be variable. The issue that the end user will experience is that they cannot download Internet pages or transfer files using File Transfer Protocol (FTP) but may be able to ping successfully.

For fragmentation problem shares among all the tunneling protocols, this is not unique to DS-Lite. The IPv4 packet isn't over-sized, it is the v6 encapsulation that MAY cause the oversized issue. So the tunnel points are responsible to handle the fragmentation. In general, the Tunnel-Entry Point and Tunnel-Exist Point should fragment and reassemble the oversize datagram. This mechanism is

transport protocol agnostic and work for both UDP and TCP. For TCP, we could potentially avoid fragmentation by modify MSS option. The B4 networking component may send an ICMP Destination Unreachable-Fragmentation Needed and DF Set message back to the sending host in the subscriber network.

2.2. Lawful Intercept Considerations

Because of its IPv4-in-IPv6 tunneling scheme, interception in DS-Lite architecture must be performed on the AFTR itself. Timestamped logging of the address and port mappings at the AFTR must be maintained, which in turn can add a heavy resource burden to the AFTR devices.

Logging to a storage device off the AFTR may also contribute to network load. Wiretapping of a single subject may mean statically mapping the user to a certain range of ports on a single address, to remove the need to follow dynamic port mappings. A single IPv4 address, or some range of ports for each address, might be set aside for wiretapping purposes to simplify such procedures. But any requirement that users behind a given AFTR be logged is going to mean logging not only traffic but all changes to the mapping tables.

2.3. Logging at the AFTR

The timestamped logging of address and port mappings is essential not only for lawful intercept but also for tracing back specific users when a problem is identified from the outside of the AFTR. Such a problem is usually a misbehaving user in the case of a spammer or a DoS source, or someone violating a usage policy. Knowing the user may result in black-listing. Without time-specific logs of the address and port mappings, a misbehaving user stays well hidden behind the AFTR.

Blacklisting might restrict others in the home or office from accessing the website but altogether few innocent bystanders are affected. What happens, though, if a website bans an IPv4 address on the outside of an AFTR? In the effort to restrict a single user, hundreds of people may be inadvertently restricted generally all subscribers on a CMTS or a group of BNASEs behind the AFTR.

Black- or white-listing may need to be split in an AFTR architecture. Policies applying to incoming sources must be implemented on the outside of the AFTR. Once the packets are translated, they cannot be easily identified by IPv4 address without some correlation with the AFTR mapping table.

2.3.1. AFTR's Policies

Policies applying on the NAT-ed addresses must be implemented on the external interface of the AFTR. Once the packets are translated, they cannot be easily identified by IPv4 address without some correlation with the AFTR mapping table. Policies applying to outgoing sources must be implemented on the customer-facing side of the AFTR for the same reason. In order to be able to deploy different services offers, multiple set of policies (e.g. QoS and ACL settings) can be configured on the AFTR: each set of policies can then be applied to a different logical tunnel interface on the AFTR.

2.4. AFTR Impacts on Internal Accounting Systems

Single points of failure, potential address pool depletion attacks, performance and scalability, effects on fragmented packets, effects on asymmetric traffic flows, required modifications to provisioning systems, required modifications to internal accounting systems.

2.4.1. AFTR Impacts on Accounting Process in Broadband Access

DS-Lite introduces challenges to IPv4 accounting process. In a typical DSL/Broadband access scenario where the Residential Gateway (RG) is acting as a B4 element, the BNAS is the IPv6 edge router which connects to the AFTR. The BNAS is normally responsible for IPv6 accounting and all the subscriber manager functions such as authentication, authorization and accounting. However, given the fact that IPv4 traffic is encapsulated into an IPv6 packet at the B4 level and only decapsulated at the AFTR level, the BNAS can't do the IPv4 accounting without examining the inner packet. AFTR is the next logical place to perform IPv4 accounting, but it will potentially introduce some additional complexity because the AFTR does not have detailed customer identity information.

The accounting process at the AFTR level is only necessary if the Service Provider requires separate per user accounting records for IPv4 and IPv6 traffic. If the per user IPv6 accounting records, collected by the BNAS, are sufficient, the additional complexity to be able to implement IPv4 accounting at the AFTR level is not required. It is important to consider that, since the IPv4 traffic is encapsulated in IPv6 packets, the data collected by the BNAS for IPv6 traffic already contain the total amount of traffic (i.e. IPv6 plus IPv4).

Even if detailed accounting records collection for IPv4 traffic may not be required, in some scenarios it would be useful for a Service Provider, to have inside the RADIUS Accounting packet, generated by the BNAS for the IPv6 traffic, a piece of information that can be

used to identify the AFTR that is handling the IPv4 traffic for that user. This can be achieved by adding into the IPv6 accounting records the RADIUS attribute information specified in [I-D.ietf-softwire-dslite-radius-ext]

2.5. Reliability Considerations of AFTR

The service provider can use techniques to achieve high availability such as various types of clusters to ensure availability of the IPv4 service. High availability techniques include the cold standby mode. In this mode the AFTR states are not replicated from the Primary AFTR to the Backup AFTR. When the Primary AFTR fails, all the existing established sessions will be flushed out. The internal hosts are required to re-establish sessions to the external hosts. Another high availability option is the hot standby mode. In this mode the AFTR keeps established sessions while failover happens. AFTR states are replicated from the Primary AFTR to the Backup AFTR. When the Primary AFTR fails, the Backup AFTR will take over all the existing established sessions. In this mode the internal hosts are not required to re-establish sessions to the external hosts. The final option is to deploy a mode in between these two whereby only selected sessions such as critical protocols are replicated. Criteria for sessions to be replicated on the backup would be explicitly configured on the AFTR devices of a redundancy group.

2.6. Strategic Placement of AFTR

The public IPv4 addresses are pulled away from the customer edge to the outside of the centralized AFTR where many customer networks can share a single public IPv4 address.

The AFTR architecture design, then, is mostly figuring out the strategic placement of each AFTR to best use the capacity of each public IPv4 address without oversubscribing the address or overtaxing the AFTR itself. Although only a few studies of per-user port usage have been done, an AFTR should be able to support 3000 - 5000 users per public IPv4 address.

By centralizing public IPv4 addresses, each address no longer represents a single machine, a single household, or a single small office. The address now represents thousands of machines, homes, and offices related only in that they are behind the same AFTR. Identification by IP address becomes difficult or impossible and thus applications that assume such geographic information may not work as intended.

2.7. AFTR Considerations for Geographically Aware Services

Various applications and services will place their servers in such a way to locate them near sets of user so that this will lessen the latency on the client end. In addition, having sufficient geographical coverage can indirectly improve end-to-end latency. An example is that nameservers typically return results optimized for the DNS resolver's location. Deployment of AFTR must be done in such a way as not to negatively impact the geographical nature of these services. This can be done by making sure that AFTR deployments are geographically distributed so that existing assumptions of the clients source IP address by geographically aware servers can be maintained.

2.8. Impacts on QoS

As with tunneling in general there are challenges with deep packet inspection with DS-Lite for purposes of QoS. Service Providers commonly uses DSCP to classify and prioritize packets. It is recommended the AFTR to copy the DSCP value in the IPv4 header to the IPv6 header after the encapsulation.

2.9. Port Forwarding Considerations

Some applications require accepting incoming UDP or TCP traffic. When the remote host is on IPv4, the incoming traffic will be directed towards an IPv4 address. Some applications use (UPnP-IGD) (e.g., Xbox) or ICE [I-D.ietf-mmusic-ice] (e.g., SIP, Yahoo!, Google, Microsoft chat networks), other applications have all but completely abandoned incoming connections (e.g., most FTP transfers use passive mode). But some applications rely on ALGs, UPnP IGD, or manual port configuration. Port Control Protocol (PCP) [I-D.wing-pcp-design-considerations] is designed to address this issues.

3. B4 Deployment Considerations

In order to configure the IPv4-in-IPv6 tunnel, the B4 element needs the IPv6 address of the AFTR element. This IPv6 address can be configured using a variety of methods, ranging from an out-of-band mechanism, manual configuration or a variety of DHCPv6 options. In order to guarantee interoperability, a B4 element SHOULD implement the DHCPv6 option defined in [I-D.ietf-softwire-ds-lite-tunnel-option]. The DHCP server must be reachable via normal DHCP request channels from the B4, and it must be configured with the AFTR address. In Broadband Access scenario where AAA/RADIUS is used for provisioning user profiles in the BNAS,

[I-D.ietf-softwire-dslite-radius-ext] may be used. BNAS will learn the AFTR address from the RADIUS attribute and act as the DHCPv6 server for the B4s.

3.1. DNS deployment Considerations

[I-D.ietf-softwire-dual-stack-lite] recommends configuring the B4 with a DNS proxy resolver, which will forward queries to an external recursive resolver over IPv6. Alternately, the B4 proxy resolver can be statically configured with the IPv4 address of an external recursive resolver. In this case, DNS traffic to the external resolver will be tunneled through IPv6 to the AFTR. Note that the B4 must also be statically configured with an IPv4 address in order to source packets; the draft recommends an address in the 192.0.0.0/29 range. Even more simply, you could eliminate the DNS proxy, and configure the DHCP server on the B4 to give its clients the IPv4 address of an external recursive resolver. Because of the extra traffic through the AFTR, and because of the need to statically configure the B4, these alternate solutions are likely to be unsatisfactory in a production environment. However, they may be desirable in a testing or demonstration environment.

4. Security Considerations

This document does not present any new security issues. [I-D.ietf-softwire-dual-stack-lite] discusses DS-Lite related security issues. General NAT security issues are not repeated here.

Some of the security issues with carrier-grade NAT result directly from the sharing of the routable IPv4 address. Addresses and timestamps are often used to identify a particular user, but with shared addresses, more information (i.e., protocol and port numbers) is needed. This impacts software used for logging and tracing spam, denial of service attacks, and other abuses. Devices on the customers side may try to carry out general attacks against systems on the global Internet or against other customers by using inappropriate IPv4 source addresses inside tunneled traffic. The AFTR needs to protect against such abuse. One customer may try to carry out a denial of service attack against other customers by monopolizing the available port numbers. The AFTR needs to ensure equitable access. At a more sophisticated level, a customer may try to attack specific ports used by other customers. This may be more difficult to detect and to mitigate without a complete system for authentication by port number, which would represent a huge security requirement.

5. Conclusion

DS-Lite provides new functionality to transition IPv4 traffic to IPv6 addresses. As the supply of unique IPv4 addresses diminishes, service providers can now allocate new subscriber homes IPv6 addresses and IPv6-capable equipment. DS-Lite provides a means for the private IPv4 addresses behind the IPv6 equipment to reach the IPv4 network.

This document discusses the issues that arise when deploying DS-Lite in various deployment modes. Hence, this document can be a useful reference for service providers and network designers. Deployment considerations of the B4, AFTR and DNS have been discussed and recommendations for their usage have been documented.

6. Acknowledgement

TBD

7. IANA Considerations

This memo includes no request to IANA.

8. References

8.1. Normative References

[I-D.ietf-softwire-ds-lite-tunnel-option]

Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual- Stack Lite", draft-ietf-softwire-ds-lite-tunnel-option-05 (work in progress), September 2010.

[I-D.ietf-softwire-dslite-radius-ext]

Maglione, R. and A. Durand, "RADIUS Extensions for Dual- Stack Lite", draft-ietf-softwire-dslite-radius-ext-00 (work in progress), October 2010.

[I-D.ietf-softwire-dual-stack-lite]

Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual- Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-06 (work in progress), August 2010.

[I-D.wing-pcp-design-considerations]

Wing, D., "PCP Design Considerations",
draft-wing-pcp-design-considerations-00 (work in
progress), September 2010.

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

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

8.2. Informative References

[I-D.ietf-mmusic-ice]
Rosenberg, J., "Interactive Connectivity Establishment
(ICE): A Protocol for Network Address Translator (NAT)
Traversal for Offer/Answer Protocols",
draft-ietf-mmusic-ice-19 (work in progress), October 2007.

[I-D.ietf-v6ops-ipv6-cpe-router]
Singh, H., Beebe, W., Donley, C., Stark, B., and O.
Troan, "Basic Requirements for IPv6 Customer Edge
Routers", draft-ietf-v6ops-ipv6-cpe-router-07 (work in
progress), August 2010.

[I-D.xu-behave-stateful-nat-standby]
Xu, X., "Redundancy and Load Balancing Framework for
Stateful Network Address Translators (NAT)",
draft-xu-behave-stateful-nat-standby-05 (work in
progress), September 2010.

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

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

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

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

Authors' Addresses

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

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

Roberta Maglione
Telecom Italia
Via Reiss Romoli 274
Torino 10148
Italy

Email: roberta.maglione@telecomitalia.it
URI: <http://www.telecomitalia.it>

Carl Williams
MCSR Labs
Philadelphia
U.S.A.

Email: carlw@mcsr-labs.org

Christian Jacquenet
France Telecom
Rennes
France

Email: christian.jacquenet@orange-ftgroup.com>

Mohamed Boucadair
France Telecom
Rennes
France

Email: mohamed.boucadair@orange-ftgroup.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 20, 2011

N. Matsuhira
Fujitsu Limited
October 17, 2010

Stateless Automatic IPv4 over IPv6 Tunneling with IPv4 Address Sharing
draft-matsuhira-sa46t-as-00

Abstract

This document specifies Stateless Automatic IPv4 over IPv6 Tunneling with IPv4 Address Sharing (SA46T-AS) base specification. SA46T-AS is basically the same technology with SA46T, however that have IPv4 address sharing capability. SA46T-SA is gateway technology, not protocol.

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on April 20, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Architecture of SA46T-AS	3
3. SA46T-AS address format	5
4. Using SA46T-AS in client server environments	5
4.1. Client environments	5
4.2. Server environments	6
5. Characteristic	7
6. IANA Considerations	8
7. Security Considerations	8
8. References	8
8.1. Normative References	8
8.2. References	8
Author's Address	8

1. Introduction

This document provides Stateless Automatic IPv4 over IPv6 Tunneling with IPv4 Address Sharing (SA46T-AS) base specification.

SA46T-AS is basically the same technology with SA46T[I-D.draft-matsuhira-sa46t-spec] , however that have IPv4 address sharing capability.

The basic architecture of the SA46T-AS is the same with SA46T, so SA46T-AS can provide all of SA46T function, such as making backbone network IPv6 only , or provide many IPv4 network planes over single IPv6 backbone network.

SA46T-AS add IPv4 address sharing function to SA46T. So, SA46T-AS enable many host to share single IPv4 global address.

SA46T is gateway technology, not protocol.

2. Architecture of SA46T-AS

Figure 1 shows SA46T address architecture. SA46T map IPv4 address to SA46T address keeping locator - identifier relation. The n bits identifier part of IPv4 address and n bits identifier part of IPv6 address is the same value, and the 32-n bits locator part in IPv4 address and 128 - n bits locator part in IPv6 address is the same meaning. So, the meaning of routing information is the same between IPv4 space and IPv6 space.

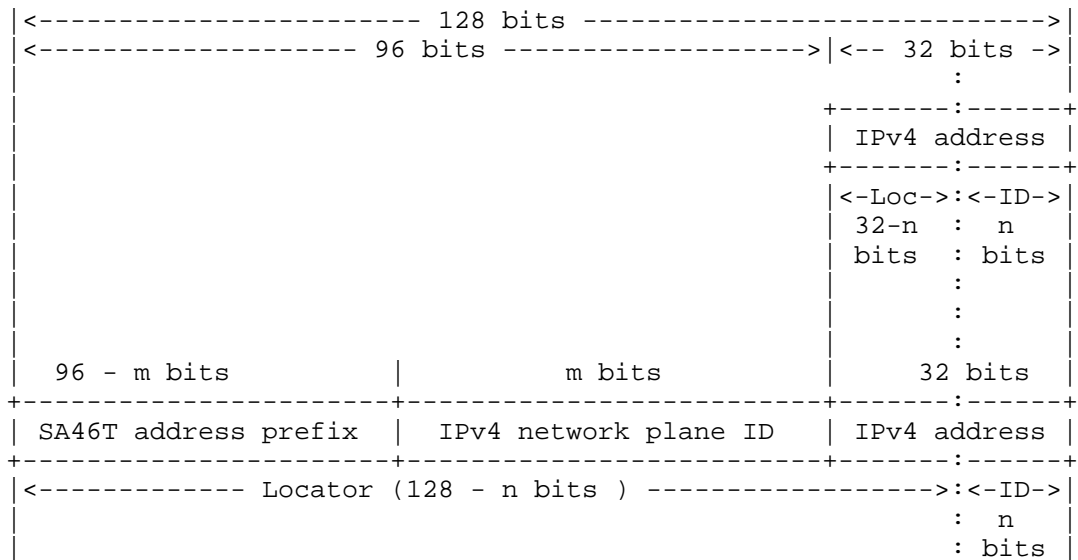


Figure 1

Figure 2 shows SA46T-AS address architecture. SA46T-AS address consists four parts, SA46T-AS prefix, IPv4 network plane ID, IPv4 address, and Port number. That mean SA46T-AS address consists SA46T address and port number.

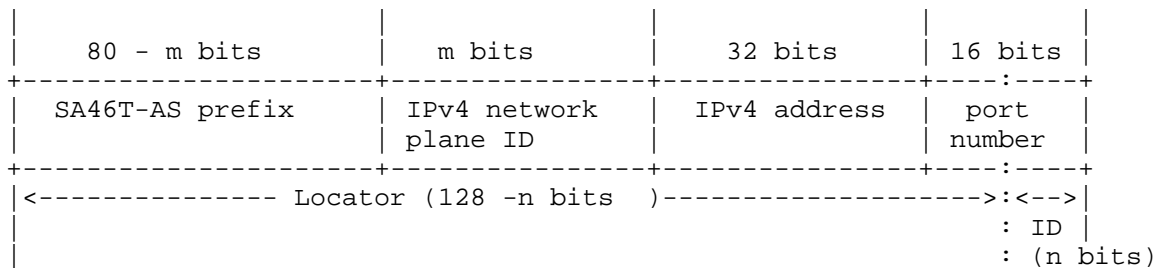


Figure 2

In SA46T, boundary of locator and identifier is in IPv4 address part, however in SA46T-AS, boundary of locator and identifier is in port number part, that mean, SA46T-AS use upper part of port number as locator, and lower part of port number as identifier.

3. SA46T-AS address format

Figure 3 show a example of SA46T-AS address format. In this example, 16bits IPv4 network plane ID is used, that provide 65535 IPv4 network plane.

```

| 3 |          45bits          | 16bits | 16 bits | 32bits | 16 bits |
+---+-----+-----+-----+-----+-----+
|001|Global routing prefix|subnet id| plane ID|IPv4 address| Port # |
+---+-----+-----+-----+-----+-----+
<---SA46T address prefix----->

```

Figure 3

4. Using SA46T-AS in client server environments

4.1. Client environments

Figure 4 shows a example of SA46T-AS usage in client environments. In this document, NAPT is IPv4 - IPv4 Netowrk address and port number translator. Coopetation with NAPT, SA46T-AS provide IPv4 address sharing with different users.

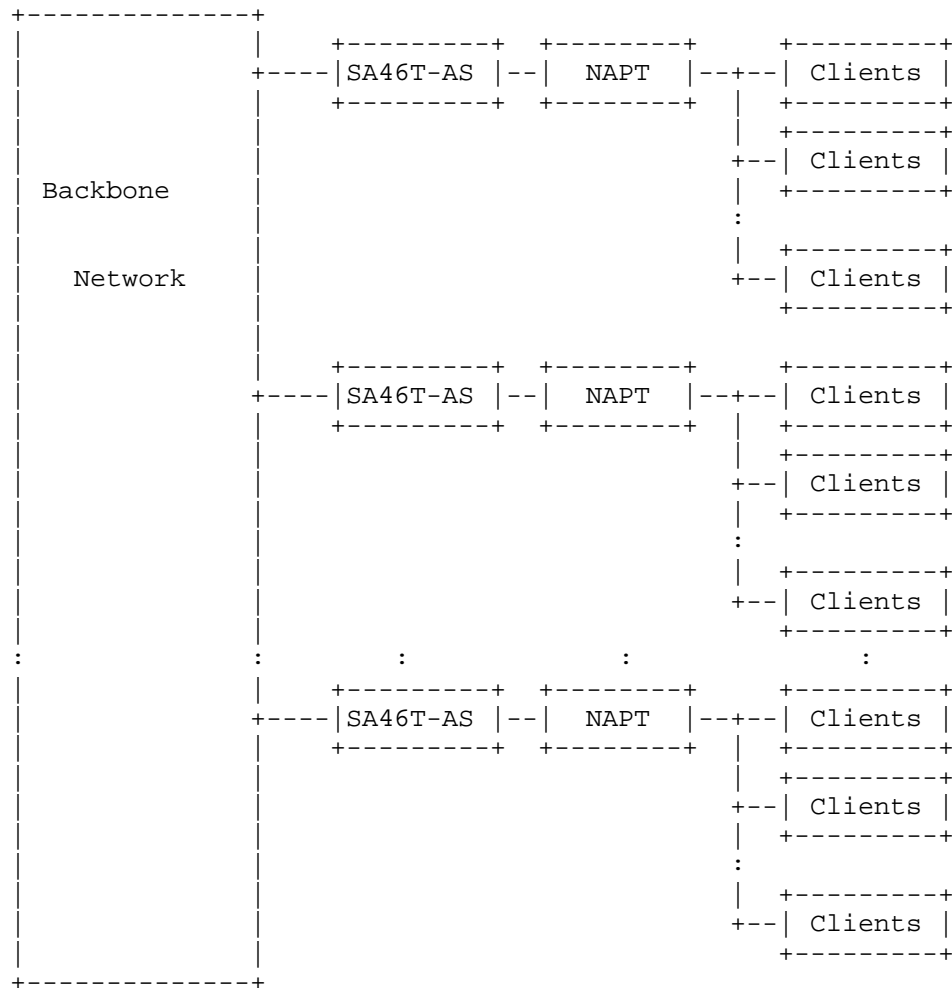


Figure 4

4.2. Server environments

Figure 5 shows an example of SA46T-AS usage in server environments. In this example, server terminate SA46T-AS tunnel. This case, Server require at least one port number per server, that mean, 128bits host route advertise for server access via IPv4. This case, full access is provided via IPv6.

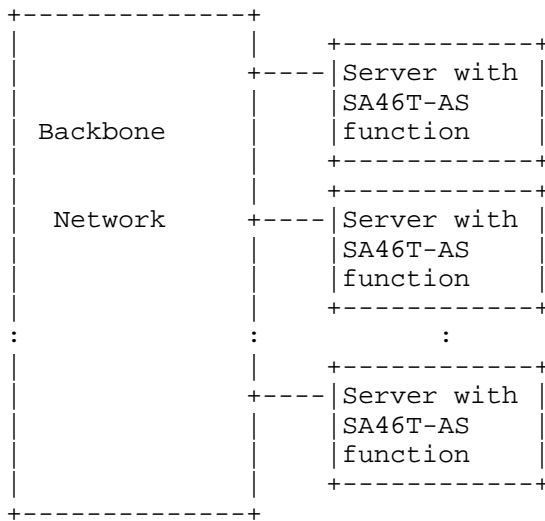


Figure 5

5. Characteristic

SA46T has following useful characteristics.

- o Reduce backbone network operation cost with IPv6 single stack (at least less than Dual Stack)
- o Can allocate IPv4 address to stub networks, which used in backbone network before installing SA46T
- o Less configuration
- o No need for special protocol
- o No dependent Layer 2 network
- o Can Stack IPv4 Private networks
- o Easy stop IPv4 operation in stub network for future (just remove SA46T)
- o Provide redundancy

Moreover, SA46T-AS add following characteristics to SA46T.

- o Provide IPv4 address sharig function

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Security Considerations

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

8. References

8.1. Normative References

- [I-D.draft-matsuhira-sa46t-spec]
Matsuhira, N., "Stateless Automatic IPv4 over IPv6 Tunneling: Specification".
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

8.2. References

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

Author's Address

Naoki Matsuhira
Fujitsu Limited
17-25, Shinkamata 1-chome, Ota-ku
Tokyo, 144-8588
Japan

Phone: +81-3-6424-6270

Fax:

Email: matsuhira@jp.fujitsu.com

Internet Research Task Force
(IRTF)
Internet-Draft
Intended status: Experimental
Expires: April 11, 2011

F. Templin, Ed.
Boeing Research & Technology
October 8, 2010

The Internet Routing Overlay Network (IRON)
draft-templin-iron-13.txt

Abstract

Since the Internet must continue to support escalating growth due to increasing demand, it is clear that current routing architectures and operational practices must be updated. This document proposes an Internet Routing Overlay Network (IRON) that supports sustainable growth through Provider Independent addressing while requiring no changes to end systems and no changes to the existing routing system. IRON further addresses other important issues including routing scaling, mobility management, multihoming, traffic engineering and NAT traversal. While business considerations are an important determining factor for widespread adoption, they are out of scope for this document. This document is a product of the IRTF Routing Research Group.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 11, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

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

Table of Contents

1. Introduction	4
2. Terminology	5
3. The Internet Routing Overlay Network	7
3.1. IRON Client Router	9
3.2. IRON Serving Router	10
3.3. IRON Relay Router	10
4. IRON Organizational Principles	11
5. IRON Initialization	13
5.1. IRON Relay Router Initialization	13
5.2. IRON Serving Router Initialization	14
5.3. IRON Client Router Initialization	15
6. IRON Operation	16
6.1. IRON Client Router Operation	16
6.2. IRON Serving Router Operation	17
6.3. IRON Relay Router Operation	18
6.4. IRON Reference Operating Scenarios	19
6.4.1. Both Hosts Within IRON EUNs	19
6.4.2. Mixed IRON and Non-IRON Hosts	22
6.5. Mobility, Multihoming and Traffic Engineering Considerations	25
6.5.1. Mobility Management	25
6.5.2. Multihoming	26
6.5.3. Inbound Traffic Engineering	26
6.5.4. Outbound Traffic Engineering	26
6.6. Renumbering Considerations	26
6.7. NAT Traversal Considerations	27
6.8. Nested EUN Considerations	27
6.8.1. Host A Sends Packets to Host Z	28
6.8.2. Host Z Sends Packets to Host A	29
7. Additional Considerations	30
8. Related Initiatives	30
9. IANA Considerations	30
10. Security Considerations	31
11. Acknowledgements	31
12. References	31
12.1. Normative References	31
12.2. Informative References	32
Appendix A. IRON VPs Over Internetworks with Different Address Families	34
Appendix B. Scaling Considerations	35
Author's Address	36

1. Introduction

Growth in the number of entries instantiated in the Internet routing system has led to concerns for unsustainable routing scaling [I-D.narten-radir-problem-statement]. Operational practices such as increased use of multihoming with IPv4 Provider-Independent (PI) addressing are resulting in more and more fine-grained prefixes injected into the routing system from more and more end-user networks. Furthermore, the forthcoming depletion of the public IPv4 address space has raised concerns for both increased address space fragmentation (leading to yet further routing table entries) and an impending address space run-out scenario. At the same time, the IPv6 routing system is beginning to see growth in IPv6 Provider-Aggregated (PA) prefixes [BGPMON] which must be managed in order to avoid the same routing scaling issues the IPv4 Internet now faces. Since the Internet must continue to scale to accommodate increasing demand, it is clear that new routing methodologies and operational practices are needed.

Several related works have investigated routing scaling issues. Virtual Aggregation (VA) [I-D.ietf-grow-va] and Aggregation in Increasing Scopes (AIS) [I-D.zhang-evolution] are global routing proposals that introduce routing overlays with Virtual Prefixes (VPs) to reduce the number of entries required in each router's Forwarding Information Base (FIB) and Routing Information Base (RIB). Routing and Addressing in Networks with Global Enterprise Recursion (RANGER) [RFC5720] examines recursive arrangements of enterprise networks that can apply to a very broad set of use case scenarios [I-D.russert-rangers]. In particular, RANGER supports encapsulation and secure redirection by treating each layer in the recursive hierarchy as a virtual non-broadcast, multiple access (NBMA) "link". RANGER is an architectural framework that includes Virtual Enterprise Traversal (VET) [I-D.templin-intarea-vet] and the Subnetwork Adaptation and Encapsulation Layer (SEAL) [I-D.templin-intarea-seal] as its functional building blocks.

This document proposes an Internet Routing Overlay Network (IRON) with goals of supporting sustainable growth while requiring no changes to the existing routing system. IRON borrows concepts from VA, AIS and RANGER, and further borrows concepts from the Internet Vastly Improved Plumbing (Ivip) [I-D.whittle-ivip-arch] architecture proposal along with its associated Translating Tunnel Router (TTR) mobility extensions [TTRMOB]. Indeed, the TTR model to a great degree inspired the IRON mobility architecture design discussed in this document. The Network Address Translator (NAT) traversal techniques adapted for IRON were inspired by the Simple Address Mapping for Premises Legacy Equipment (SAMPLE) proposal [I-D.carpenter-softwire-sample].

IRON specifically seeks to provide scalable PI addressing without changing the current BGP [RFC4271] routing system. IRON observes the Internet Protocol standards [RFC0791][RFC2460]. Other network layer protocols that can be encapsulated within IP packets (e.g., OSI/CLNP [RFC1070], etc.) are also within scope.

The IRON is a global routing system comprising virtual overlay networks managed by Virtual Prefix Companies (VPCs) that own and manage Virtual Prefixes (VPs) from which End User Network (EUN) PI prefixes (EPs) are delegated to customer sites. The IRON is motivated by a growing customer demand for multihoming, mobility management and traffic engineering while using stable PI addressing to avoid network renumbering [RFC4192][RFC5887]. The IRON uses the existing IPv4 and IPv6 global Internet routing systems as virtual links for tunneling inner network protocol packets within outer IPv4 or IPv6 headers (see: Section 3). The IRON requires deployment of a small number of new BGP core routers and supporting servers, as well as IRON-aware routers/servers in customer EUNs. No modifications to hosts, and no modifications to most routers are required.

While the IRON architecture addresses network mobility, host mobility considerations are outside the scope of this document. IP multicast considerations are also out of scope.

Note: This document is offered in compliance with Internet Research Task Force (IRTF) document stream procedures [RFC5743]; it is not an IETF product and is not a standard. The views in this document were considered controversial by the IRTF Routing Research Group (RRG) but the RG reached a consensus that the document should still be published. The document will undergo a period of review within the RRG and through selected expert reviewers prior to publication. The following sections discuss details of the IRON architecture.

2. Terminology

This document makes use of the following terms:

End User Network (EUN)

an edge network that connects an organization's devices (e.g., computers, routers, printers, etc.) to the Internet.

End User Network PI Prefix (EP)

a more-specific Provider-Independent (PI) prefix derived from a Virtual Prefix (VP) (e.g., an IPv4 /28, an IPv6 /56, etc.) and delegated to an EUN by a Virtual Prefix Company (VPC).

End User Network PI Address (EPA)

a network layer address belonging to an EP and assigned to the interface of an end system in an EUN.

Forwarding Information Based (FIB)

a data structure containing network prefix to next-hop mappings; usually maintained in a router's fast-path processing lookup tables.

Internet Routing Overlay Network (IRON)

a composite virtual overlay network that comprises the union of all VPC overlay networks configured over a common Internetwork. The IRON supports routing through encapsulation of inner packets with EPA addresses within outer headers that use locator addresses.

IRON Client Router ("Client")

a customer's router (or host with embedded gateway function) that logically connects the customer's EUNs and their associated EPs to the IRON via tunnels.

IRON Serving Router ("Server")

a VPC's overlay network router that provides forwarding and mapping services for the EPs owned by customer Client routers.

IRON Relay Router ("Relay")

a VPC's overlay network router that acts as a relay between the IRON and the native Internet.

IRON Router (IR)

generically refers to any of an IRON Client/Server/Relay router.

Internet Service Provider (ISP)

a service provider which connects customer EUNs to the underlying Internetwork. In other words, an ISP is responsible for providing basic Internet connectivity for customer EUNs.

Locator

an IP address assigned to the interface of a router or end system within a public or private network. Locators taken from public IP prefixes are routable on a global basis, while locators taken from private IP prefixes are made public via Network Address Translation (NAT).

Provider Aggregated (PA) address or prefix

a network layer address or prefix delegated to an EUN by an ISP.

Provider Independent (PI) address or prefix

a network layer address or prefix delegated to an EUN by a third party independently of the EUN's ISP arrangements.

Routing and Addressing in Networks with Global Enterprise Recursion (RANGER)

an architectural examination of virtual overlay networks applied to enterprise network scenarios, with implications for a wider variety of use cases.

Subnetwork Encapsulation and Adaptation Layer (SEAL)

an encapsulation sublayer that provides extended packet identification and a control message protocol to ensure deterministic network-layer feedback.

Virtual Enterprise Traversal (VET)

a method for discovering border routers and forming dynamic point-to-(multi)point tunnels over enterprise networks (or sites) with varying properties.

Virtual Prefix (VP)

a PI prefix block (e.g., an IPv4 /16, an IPv6 /20, an OSI NSAP prefix, etc.) that is owned and managed by a Virtual Prefix Company (VPC).

Virtual Prefix Company (VPC)

a company that owns and manages a set of VPs from which it delegates EPs to EUNs.

VPC Overlay Network

a specialized set of routers deployed by a VPC to service customer EUNs through a virtual overlay network configured over an underlying Internetwork (e.g., the global Internet).

3. The Internet Routing Overlay Network

The Internet Routing Overlay Network (IRON) is a system of virtual overlay networks configured over a common Internetwork. While the principles presented in this document are discussed within the context of the public global Internet, they can also be applied to any autonomous Internetwork. The rest of this document therefore refers to the terms "Internet" and "Internetwork" interchangeably except in cases where specific distinctions must be made.

The IRON consists of IRON Routers (IRs) that automatically tunnel the packets of end-to-end communication sessions within encapsulating headers used for Internet routing. IRs use Virtual Enterprise

Traversal (VET) [I-D.templin-intarea-vet] in conjunction with the Subnetwork Encapsulation and Adaptation Layer (SEAL) [I-D.templin-intarea-seal] to encapsulate inner network layer packets within outer headers as shown in Figure 1:

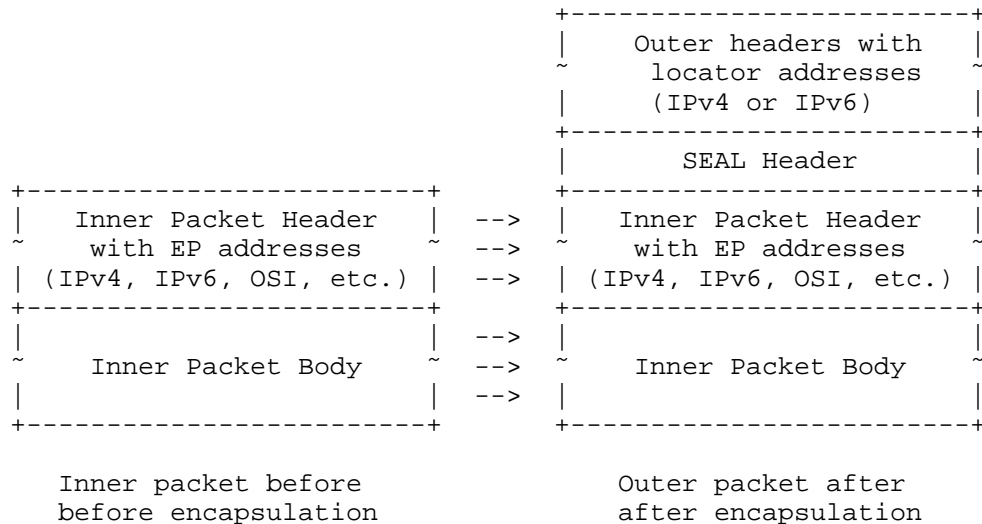


Figure 1: Encapsulation of Inner Packets Within Outer IP Headers

VET specifies the automatic tunneling mechanisms used for encapsulation, while SEAL specifies the format and usage of the SEAL header as well as a set of control messages. Most notably, IRs use the SEAL Control Message Protocol (SCMP) to deterministically exchange and authenticate control messages such as route redirections, indications of Path Maximum Transmission Unit (PMTU) limitations, destination unreachable, etc.

The IRON is the union of all virtual overlay networks that are configured over a common underlying Internet and are owned and managed Virtual Prefix Companies (VPCs). Each such virtual overlay network comprises a set of IRs distributed throughout the Internet to serve highly-aggregated Virtual Prefixes (VPs). VPCs delegate sub-prefixes from their VPs which they lease to customers as End User Network PI prefixes (EPs). The customers in turn assign the EPs to their customer edge IRs which connect their End User Networks (EUNs) to the IRON.

VPCs may have no affiliation with the ISP networks from which customers obtain their basic Internet connectivity. Therefore, a customer could procure its summary network services either through a common broker or through separate entities. In that case, the VPC

can open for business and begin serving its customers immediately without the need to coordinate its activities with ISPs or with other VPCs. Further details on business considerations are out of scope for this document.

The IRON requires no changes to end systems and no changes to most routers in the Internet. Instead, the IRON comprises IRs that are deployed either as new platforms or as modifications to existing platforms. IRs may be deployed incrementally without disturbing the existing Internet routing system, and act as waypoints (or "cairns") for navigating the IRON. The functional roles for IRs are described in the following sections.

3.1. IRON Client Router

An IRON client router (or, simply, "Client") is a customer's router (or host with embedded gateway function) that logically connects the customer's EUNs and their associated EPs to the IRON via tunnels as shown in Figure 2. Clients obtain EPs from VPCs and use them to number subnets and interfaces within their EUNs. A Client can be deployed on the same physical platform that also connects the customer's EUNs to its ISPs, but it may also be a separate router or even a standalone server system located within the EUN. (This model applies even if the EUN connects to the ISP via a Network Address Translator (NAT) - see Section 6.7).

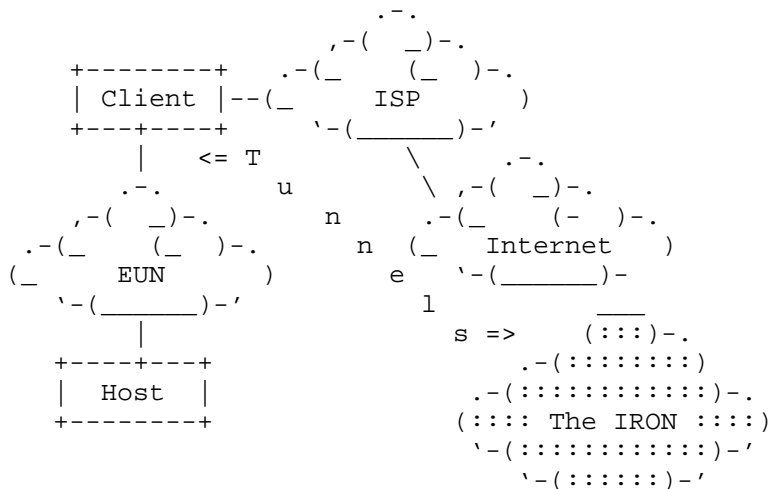


Figure 2: IRON Client Router Connecting EUN to the IRON

3.2. IRON Serving Router

An IRON serving router (or, simply, "Server") is a VPC's overlay network router that provides forwarding and mapping services for the EPs owned by customer Client routers. In typical deployments, a VPC will deploy many Servers around the IRON in a globally-distributed fashion (e.g., as depicted in Figure 3) so that Clients can discover those that are nearby.

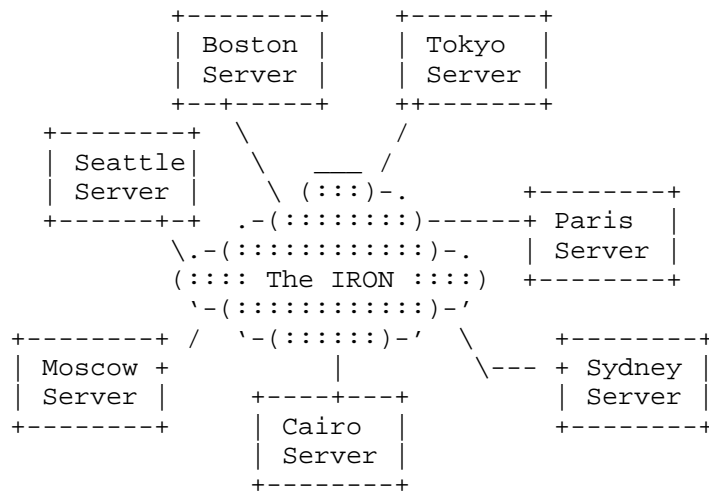


Figure 3: IRON Serving Router Global Distribution Example

Each Server acts as tunnel-endpoint router that forms a bi-directional tunnel with each of its Client customers. Each Server also associates with a set of Relays that can forward packets from the IRON out to the native Internet and vice-versa as discussed in the next section.

3.3. IRON Relay Router

An IRON Relay Router (or, simply, "Relay") is a VPC's overlay network router that acts as a relay between the IRON and the native Internet. It therefore also serves as an Autonomous System Border Router (ASBR) that is owned and managed by the VPC.

Each VPC configures one or more Relays which advertise the company's VPs into the IPv4 and IPv6 global Internet BGP routing systems. Each Relay associates with all of the VPC's overlay network Servers, e.g., via tunnels over the IRON, via a direct interconnect such as an Ethernet cable, etc. The Relay role (as well as its relationship with overlay network Servers) is depicted in Figure 4:

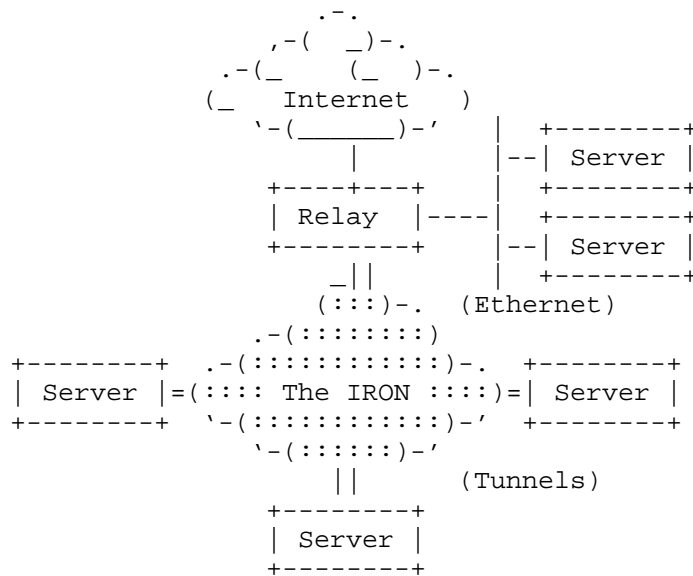


Figure 4: IRON Relay Router Connecting IRON to Native Internet

4. IRON Organizational Principles

The IRON consists of the union of all VPC overlay networks configured over a common Internetwork (e.g., the public Internet). Each such overlay network represents a distinct "patch" on the Internet "quilt", where the patches are stitched together by tunnels over the links, routers, bridges, etc., that connect the underlying. When a new VPC overlay network is deployed, it becomes yet another patch on the quilt. The IRON is therefore a composite overlay network consisting of multiple individual patches, where each patch coordinates its activities independently of all others (with the exception that the Servers of each patch must be aware of all VPs in the IRON). In order to ensure mutual cooperation between all VPC overlay networks, sufficient address space portions of the inner network layer protocol (e.g., IPv4, IPv6, etc.) should be set aside and designated as VP space.

Each VPC overlay network in the IRON maintains a set of Relays and Servers that provide services to their Client customers. In order to ensure adequate customer service levels, the VPC should conduct a traffic scaling analysis and distribute sufficient Relays and Servers for the overlay network globally throughout the Internet. Figure 5 depicts the logical arrangement of Relays Servers and Clients in an IRON virtual overlay network:

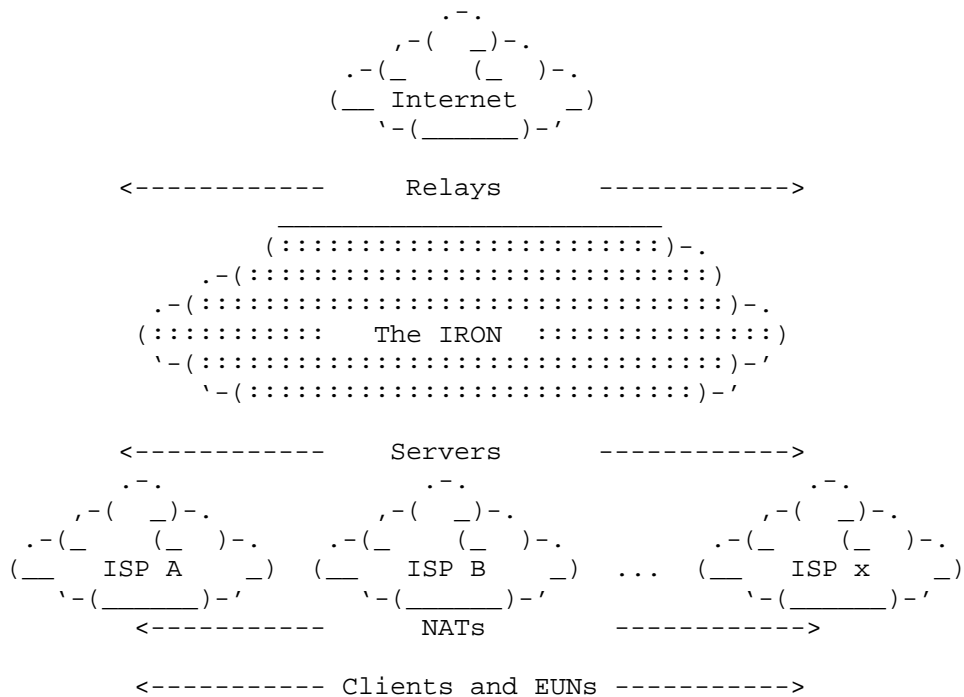


Figure 5: Virtual Overlay Network Organization

Each Relay in the VPC overlay network connects the overlay directly to the underlying IPv4 and IPv6 Internets. It also advertises the VPC overlay network's IPv4 VPs into the IPv4 BGP routing system and advertises the overlay network's IPv6 VPs into the IPv6 BGP routing system. Relays will therefore receive packets with EPA destination addresses sent by end systems in the Internet and direct them toward EPA-addressed end systems connected to the VPC overlay network.

Each VPC overlay network also manages a set of Servers that connect their Clients and associated EUNs to the IRON and to the IPv6 and IPv4 Internets via their associations with Relays. IRON Servers therefore need not be BGP routers themselves and can be simple commodity hardware platforms. Moreover, the Server and Relay functions can be deployed together on the same physical platform as a unified gateway or they may be deployed on separate platforms (e.g., for load balancing purposes).

Each Server maintains a working set of Clients for which it caches EP-to-Client mappings in its Forwarding Information Base (FIB). Each Server also in turn propagates the list of EPs in its working set to each of the Relays in the VPC overlay network via a dynamic routing

protocol (e.g., an overlay network internal BGP instance that carries only the EP-to-Server mappings and does not interact with the external BGP routing system). Each Server therefore only needs to track the EPs for its current working set of Clients, while each Relay will maintain a full EP-to-Server mapping table that represents reachability information for all EPs in the VPC overlay network.

Customers establish Clients that obtain their basic Internet connectivity from ISPs and connect to Servers to attach their EUNs to the IRON. Each EUN can connect to the IRON via one or multiple Clients as long as the Clients coordinate with one another, e.g., to mitigate EUN partitions. Unlike Relays and Servers, Clients may use private addresses behind one or several layers of NATs. Each Client initially discovers a list of nearby Servers through an anycast discovery process (described below). It then selects one of these nearby Servers and forms a bidirectional tunnel through an initial exchange followed by periodic keepalives.

After the Client selects a Server, it forwards initial outbound packets from its EUNs by tunneling them to the Server which in turn forwards them to the nearest Relay within the IRON that serves the final destination. The Client will subsequently receive redirect messages informing it of a more direct route through a Server that serves the final destination EUN.

The IRON can also be used to support VPs of network layer address families that cannot be routed natively in the underlying Internetwork (e.g., OSI/CLNP over the public Internet, IPv6 over IPv4-only Internetworks, IPv4 over IPv6-only Internetworks, etc.). Further details for support of IRON VPs of one address family over Internetworks based on other address families are discussed in Appendix A.

5. IRON Initialization

IRON initialization entails the startup actions of IRs within the VPC overlay network and customer EUNs. The following sections discuss these startups procedures.

5.1. IRON Relay Router Initialization

Before its first operational use, each Relay in a VPC overlay network is provisioned with the list of VPs that it will serve as well as the locators for all Servers that belong to the same overlay network. The Relay is also provisioned with external BGP interconnections the same as for any BGP router.

Upon startup, the Relay engages in BGP routing exchanges with its peers in the IPv4 and IPv6 Internets the same as for any BGP router. It then connects to all of the Servers in the overlay network (e.g., via a TCP connection over a bidirectional tunnel, via an iBGP route reflector, etc.) for the purpose of discovering EP->Server mappings. After the Relay has fully populated its EP->Server mapping information database, it is said to be "synchronized" wrt its VPs.

After this initial synchronization procedure, the Relay then advertises the overlay network's VPs externally. In particular, the Relay advertises the IPv6 VPs into the IPv6 BGP routing system and additionally advertises the IPv4 VPs into the IPv4 BGP routing system. The Relay additionally advertises an IPv4 /24 companion prefix (e.g., 192.0.2.0/24) into the IPv4 routing system and an IPv6 ::/64 companion prefix (e.g., 2001:DB8::/64) into the IPv6 routing system (note that these may also be sub-prefixes taken from a VP). The Relay then configures the host number '1' in the IPv4 companion prefix (e.g., as 192.0.2.1) and the interface identifier '0' in the IPv6 companion prefix (e.g., as 2001:DB8::0) and assigns the resulting addresses as subnet router anycast addresses [RFC3068][RFC2526] for the VPC overlay network. (See Appendix A for more information on the discovery and use of companion prefixes.) The Relay then engages in ordinary packet forwarding operations.

5.2. IRON Serving Router Initialization

Before its first operational use, each Server in a VPC overlay network is provisioned with the locators for all Relays that aggregate the overlay network's VPs. In order to support route optimization, the Server must also be provisioned with the list of all VPs in the IRON (i.e., and not just the VPs of its own overlay network) so that it can discern EPA and non-EPA addresses. (The Server could therefore be greatly simplified if the list of VPs could be covered within a small number of very short prefixes, e.g., one or a few IPv6 ::/20's). The Server must also discover the VP companion prefix relationships discussed in Section 5.1, e.g., via a global database such as discussed in Appendix A.

Upon startup, each Server must connect to all of the Relays within its overlay network (e.g., via a TCP connection over a bidirectional tunnel, via an iBGP route reflector, etc.) for the purpose of reporting its EP->Server mappings. The Server then actively listens for Client customers which register their EP prefixes as part of establishing a bidirectional tunnel. When a new Client registers its EP prefixes, the Server announces the new EP additions to all Relays; when an existing Client unregisters its EP prefixes, the Server withdraws its announcements.

5.3. IRON Client Router Initialization

Before its first operational use, each Client must obtain one or more EPs from its VPC as well as the companion prefixes associated with the VPC overlay network (see Section 5.1). The Client must also obtain a certificate and a public/private key pair from the VPC that it can later use to prove ownership of its EPs. This implies that each VPC must run its own public key infrastructure to be used only for the purpose of verifying its customers' claimed right to use an EP. Hence, the VPC need not coordinate its public key infrastructure with any other organization.

Upon startup, the Client sends an SCMP Router Solicitation (SRS) message to the VPC overlay network subnet router anycast address to discover the nearest Relay. The Relay will return an SCMP Router Advertisement message that lists the locator addresses of one or more nearby Servers. (This list is analogous to the ISATAP Potential Router List (PRL) [RFC5214].)

After the Client receives an SRA message from the nearby Relay listing the locator addresses of nearby Servers, it sends SRS test messages to one or more of the locator addresses to elicit SRA messages. The Server that configures the locator will include the header of the soliciting SRS message in its SRA message so that the Client can determine the number of hops along the forward path. The Server also includes a metric in its SRA messages indicating its service availability so that the Client can avoid selecting Servers that are overloaded. The Server also includes a challenge/response puzzle that the Client must answer if it wishes to connect to this Server.

When the Client receives these SRA messages, it can measure the round trip time between sending the SRS and receiving the SRA as an indication of round-trip delay. If the Client wishes to enlist the services of a specific Server (e.g., based on the measured performance), it then calculates the answer to the puzzle using its keying information and sends the answer back to the Server in a new SRS message that also contains all of the Client's EP prefixes for which it claims ownership. If the Client solved the puzzle correctly, the Server will send back a new SRA message that includes a non-zero default router lifetime and that signifies the establishment of a bidirectional tunnel. (A zero default router lifetime on the other hand signifies that the Server is currently unable to establish a bidirectional tunnel, e.g., due to heavy load, due to challenge/response failure, etc.)

Note that in the above procedure it is essential that the Client select one and only one Server. This is to allow the VPC overlay

network mapping system to have one and only one active EP-to-Server mapping at any point in time which shares fate with the Server itself. If this Server fails, the Client will quickly select a new one which will automatically update the VPC overlay network mapping system with a new EP-to-Server mapping.

6. IRON Operation

Following the IRON initialization detailed in Section 5, IRs engage in the steady-state process of receiving and forwarding packets. All IRs forward encapsulated packets over the IRON using the mechanisms of VET [I-D.templin-intarea-vet] and SEAL [I-D.templin-intarea-seal], while Relays (and in some cases Servers) additionally forward packets to and from the native IPv6 and IPv4 Internets. IRs also use SCMP to coordinate with other IRs, including the process of sending and receiving redirect messages, error messages, etc. (Note however that an IR must not send an SCMP message in response to an SCMP error message.) Each IR operates as specified in the following sub-sections.

6.1. IRON Client Router Operation

After selecting its Server as specified in Section 5.3, the Client should register each of its ISP connections with the Server in order to establish multiple bidirectional tunnels for multihoming purposes. To do so, it sends periodic SRS messages to its Server via each of its ISPs to establish additional bidirectional tunnels and to keep each tunnel alive. These messages need not include challenge/response mechanisms since prefix proof of ownership was already established in the initial exchange and a nonce in the SEAL header can be used to confirm that the SRS message was sent by the correct Client. This implies that a single nonce is used to represent the set of all bidirectional tunnels between the Client and the Server. Therefore, there are multiple bidirectional tunnels, and the nonce names this "bundle" of tunnels. (The Client and Server may conceptually represent this "bundle" as a single tunnel with multiple locator addresses, however each such locator address must be tested independently in case there are NATs on the path.)

If the Client ceases to receive SRA messages from its Server via a specific ISP connection, it marks the Server as unreachable from that address and therefore over that ISP connection. (The Client should also inform its Server of this outage via one of its working ISP connections.) If the Client ceases to receive SRA messages from its Server via multiple ISP connections, it marks the Server as unusable and quickly attempts to establish a bidirectional tunnel with a new Server. The act of establishing the tunnel with a new Server will

automatically purge the stale mapping state associated with the old Server.

When an end system in an EUN sends a flow of packets to a correspondent, the packets are forwarded through the EUN via normal routing until they reach the Client, which then tunnels the initial packets to its Server as the next hop. In particular, the Client encapsulates each packet in an outer header with its locator as the source address and the locator of its Server as the destination address. Note that after sending the initial packets of a flow, the Client may receive important SCMP messages such as indications of PMTU limitations, redirects that point to a better next hop, etc. It is therefore essential that the Client send the initial packets through its Server to avoid loss of SCMP messages that cannot traverse a NAT in the reverse direction. (The Server also provides a control point for inbound traffic engineering and a mobility anchor point and hence cannot be bypassed in the inbound direction).

The Client uses the mechanisms specified in VET and SEAL to encapsulate each forwarded packet. The Client further uses the SCMP protocol to coordinate with other IRs, including accepting redirects and other SCMP messages. When the Client receives an SCMP message, it checks the nonce field of the encapsulated packet-in-error to verify that the message corresponds to the tunnel to its Server and accepts the message if the nonce matches. (Note however that the outer source and destination addresses of the packet-in-error may be different than those in the original packet due to possible Server and/or Relay address rewritings.)

6.2. IRON Serving Router Operation

After the Server is initialized, it responds to SRSs from Clients by sending SRAs as described in Section 6.1. When the Server receives an SRS message from a new Client, it sends back an SRA message with a challenge/response puzzle. The Client in turn sends an SRS message with an answer to the puzzle. If this authentication fails, the Server discards the message. Otherwise, it creates tunnel state for this new Client, records the Client's EPs (see Section 5.3) in its FIB, and records the locator address from the SCMP message as the link-layer address of the next hop. The Server next sends an SRA message back to the Client to complete the tunnel establishment.

When the Server receives a SEAL-encapsulated packet from one of its Client tunnel endpoints, it examines the inner destination address. If the inner destination address is not an EPA, the Server decapsulates the packet and forwards it unencapsulated into the Internet if it is able to do so without loss due to ingress filtering. Otherwise, the Server re-encapsulates the packet (i.e.,

it removes the outer header and replaces it with a new outer header of the same address family) and sets the outer destination address to the locator address of an Relay within its VPC overlay network. It then forwards the re-encapsulated packet to the Relay, which will in turn decapsulate it and forward it into the Internet.

If the inner destination address is an EPA, however, the Server rewrites the outer source address to one of its own locator addresses and rewrites the outer destination address to the subnet router anycast address taken from the companion prefix associated with the inner destination address (where the companion prefix of the same address family as the outer IP protocol is used). The Server then forwards the revised packet into the Internet via a default or more-specific route, where it will be directed to the closest Relay within the destination VPC overlay network. After sending the packet, the Server may then receive an SCMP error or redirect message from a Relay/Server within the destination VPC overlay network. In that case, the Server verifies that the nonce in the message matches the tunnel corresponding to the Client that sent the original inner packet and discards the message if the nonce does not match. Otherwise, the Server re-encapsulates the SCMP message in a new outer header that uses the source address, destination address and nonce parameters associated with the tunnel to the Client; it then forwards the message to the Client. This arrangement is necessary to allow SCMP messages to flow through any NATs on the path.

When a Server ('A') receives a SEAL-encapsulated packet from a Relay or from the Internet, if the inner destination address matches an EP in its FIB 'A' re-encapsulates the packet in a new outer header that uses the source address, destination address and nonce parameters associated with the tunnel and forwards it to a Client ('B') which in turn decapsulates the packet and forwards it to the correct end system in the EUN. If 'B' has left notice with 'A' that it has moved to a new Server ('C'), however, 'A' will instead forward the packet to 'C' and also send an SCMP redirect message back to the source of the packet. In this way, 'B' can leave behind forwarding information when changing between Servers 'A' and 'C' (e.g., due to mobility events) without exposing packets to loss.

6.3. IRON Relay Router Operation

After each Relay has synchronized its VPs (see: Section 5.1) it advertises the full set of the company's VPs and companion prefixes into the IPv4 and IPv6 Internet BGP routing systems. These prefixes will be represented as ordinary routing information in the BGP, and any packets originating from the IPv4 or IPv6 Internet destined to an address covered by one of the prefixes will be forwarded to one of the VPC overlay network's Relays.

When a Relay receives a packet from the Internet destined to an EPA covered by one of its VPs, it behaves as an ordinary IP router. In particular, the Relay looks in its FIB to discover a locator of the Server that serves the EP that covers the destination address. The Relay then simply encapsulates the packet with its own locator as the outer source address and the locator of the Server as the outer destination address and forwards the packet to the Server.

When a Relay receives a packet from the Internet destined to one of its subnet router anycast addresses, it discards the packet if it is not SEAL-encapsulated. If the packet is an SCMP SRS message, the Relay instead sends an SRA message back to the source listing the locator addresses of nearby Servers then discards the message. The Relay otherwise discards all other SCMP messages.

If the packet is an ordinary SEAL packet (i.e., one that encapsulates an inner packet) the Relay sends an SCMP redirect message of the same address family back to the source with the locator of the Server that serves the EPA destination in the inner packet as the redirected target. The source and destination addresses of the SCMP redirect message use the outer destination and source addresses of the original packet, respectively. After sending the redirect message, the Relay then rewrites the outer destination address of the SEAL-encapsulated packet to the locator of the Server and forwards the revised packet to the Server. Note that in this arrangement any errors that occur on the path between the Relay and the Server will be delivered to the original source but with a different destination address due to this Relay address rewriting.

6.4. IRON Reference Operating Scenarios

The IRON supports communications when one or both hosts are located within EP-addressed EUNs regardless of whether the EPs are provisioned by the same VPC or by different VPCs. When both hosts are within IRON EUNs, route redirections that eliminate unnecessary Servers and Relays from the path are possible. When only one host is within an IRON EUN, however, route optimization cannot be used. The following sections discuss the two scenarios.

6.4.1. Both Hosts Within IRON EUNs

When both hosts are within IRON EUNs, it is sufficient to consider the scenario in a unidirectional fashion, i.e., by tracing packet flows only in the forward direction from the source host to destination host. The reverse direction can be considered separately, and incurs the same considerations as for the forward direction.

In this scenario, the initial packets of a flow produced by a source host within an EUN connected to the IRON by a Client must flow through both the Server of the source host and a Relay of the destination host, but route optimization can eliminate these elements from the path for subsequent packets in the flow. Figure 6 shows the flow of initial packets from host A to host B within two IRON EUNs (the same scenario applies whether the two EUNs are within the same VPC overlay network or different overlay networks):

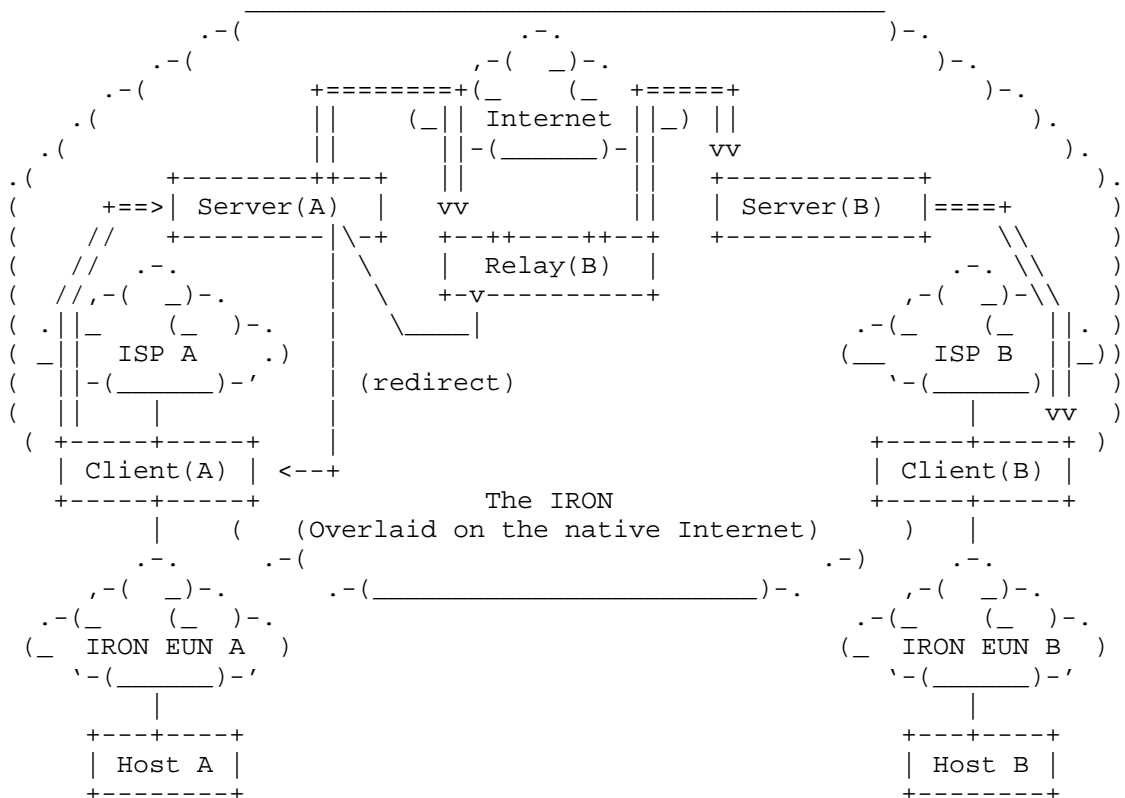


Figure 6: Initial Packet Flow Before Redirects

With reference to Figure 6, host A sends packets destined to host B via its network interface connected to EUN A. Routing within EUN A will direct the packets to Client(A) as a default router for the EUN which then uses VET and SEAL to encapsulate them in outer headers with its locator address as the outer source address and the locator address of Server(A) as the outer destination address. Client(A) then simply releases the encapsulated packets into its ISP network connection that provided its locator. The ISP will release the

packets into the Internet without filtering since the (outer) source address is topologically correct. Once the packets have been released into the Internet, routing will direct them to Server(A).

Server(A) receives the encapsulated packets from Client(A) then rewrites the outer source address to one of its own locator addresses, and rewrites the outer destination address to the subnet router anycast address of the appropriate address family associated with the inner destination address. Server(A) then releases the revised packets into the Internet where routing will direct them to Relay(B).

Relay(B) will intercept the encapsulated packets from Server(A) then check its FIB to discover an entry that covers inner destination address B with Server(B) as the next hop. Relay(B) then returns SCMP redirect messages to Server(A) (*), rewrites the outer destination address of the encapsulated packets to the locator address of Server(B), and forwards these revised packets to Server(B).

Server(B) will receive the encapsulated packets from Relay(B) then check its FIB to discover an entry that covers destination address B with Client(B) as the next hop. Server(B) then re-encapsulates the packets in a new outer header that uses the source address, destination address and nonce parameters associated with the tunnel to Client(B). Server(B) then releases these re-encapsulated packets into the Internet, where routing will direct them to Client(B). Client(B) will in turn decapsulate the packets and forward the inner packets to host B via EUN B.

(*) Note that after the initial flow of packets, Server(A) will have received one or more SCMP redirect messages from Relay(B) listing Server(B) as a better next hop. Server(A) will in turn forward the redirects to Client(A), which will thereafter forward its encapsulated packets directly to the locator address of Server(B) without involving either Server(A) or Relay(B) as shown in Figure 7:

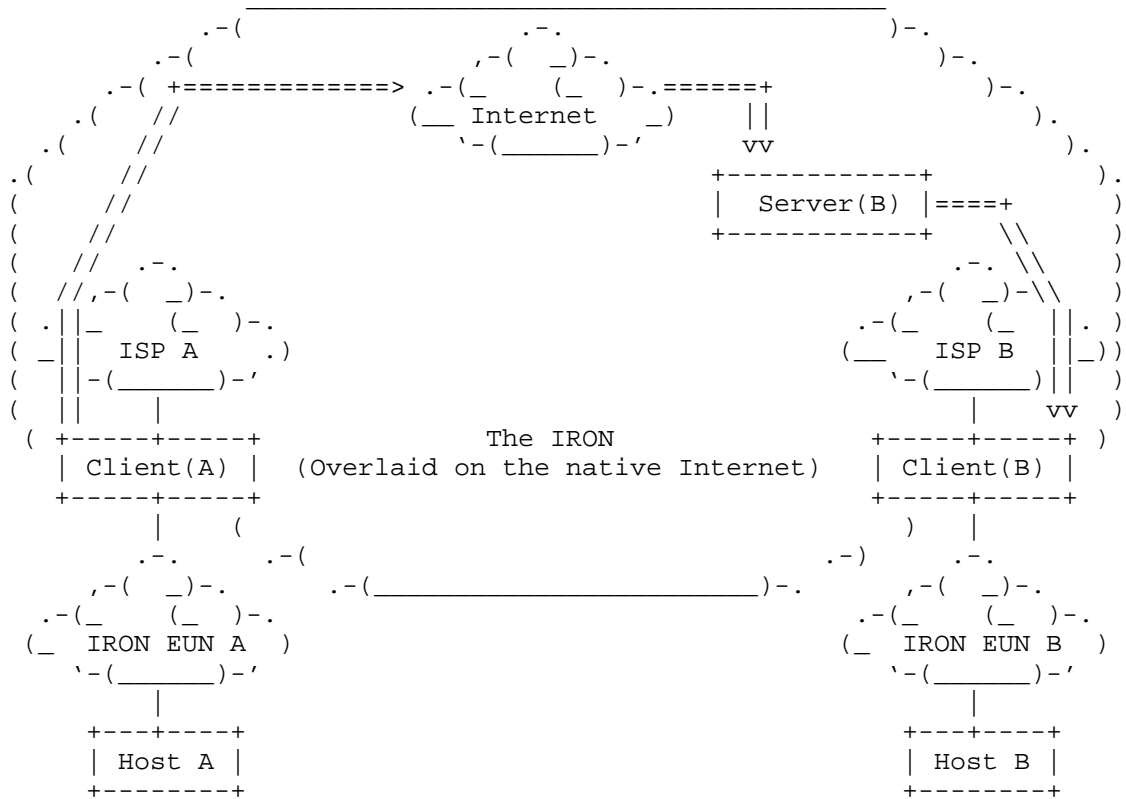


Figure 7: Sustained Packet Flow After Redirects

6.4.2. Mixed IRON and Non-IRON Hosts

When one host is within an IRON EUN and the other is in a non-IRON EUN (i.e., one that connects to the native Internet instead of the IRON), the IR elements involved depend on the packet flow directions. The cases are described in the following sections.

6.4.2.1. From IRON Host A to Non-IRON Host B

Figure 8 depicts the IRON reference operating scenario for packets flowing from Host A in an IRON EUN to Host B in a non-IRON EUN:

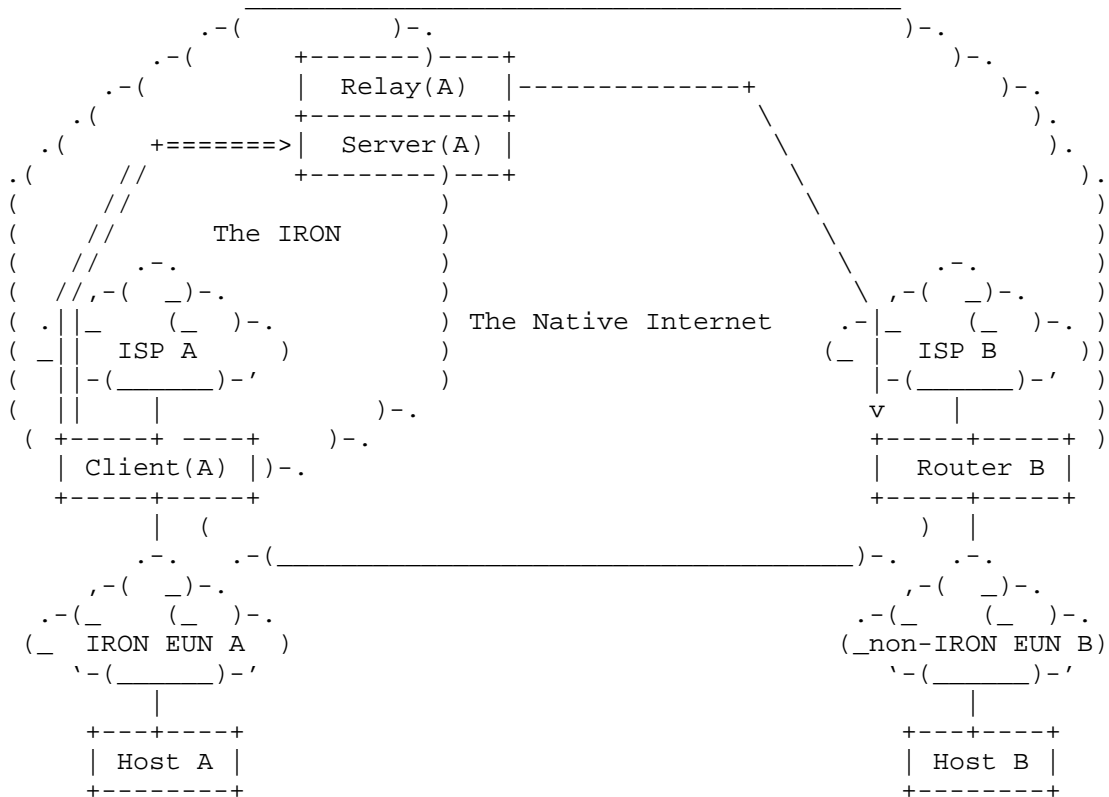


Figure 8: From IRON Host A to Non-IRON Host B

In this scenario, host A sends packets destined to host B via its network interface connected to IRON EUN A. Routing within EUN A will direct the packets to Client(A) as a default router for the EUN which then uses VET and SEAL to encapsulate them in outer headers with its locator address as the outer source address and the locator address of Server(A) as the outer destination address. The ISP will pass the packets without filtering since the (outer) source address is topologically correct. Once the packets have been released into the native Internet, routing will direct them to Server(A).

Server(A) receives the encapsulated packets from Client(A) then re-encapsulates and forwards them to Relay(A), which simply decapsulates them and releases the unencapsulated packets into the Internet. Once the packets are released into the Internet, routing will direct them to the final destination B. (Note that Server(A) and Relay(A) are depicted in Figure 8 as two halves of a unified gateway. In that case, the "forwarding" between Server(A) and Relay(A) is a zero-

instruction imaginary operation within the gateway.)

This scenario always involves a Server and Relay owned by the VPC that provides service to IRON EUN A. It therefore imparts a cost that would need to be borne by either the VPC or its customers.

6.4.2.2. From Non-IRON Host B to IRON Host A

Figure 9 depicts the IRON reference operating scenario for packets flowing from Host B in an Non-IRON EUN to Host A in an IRON EUN:

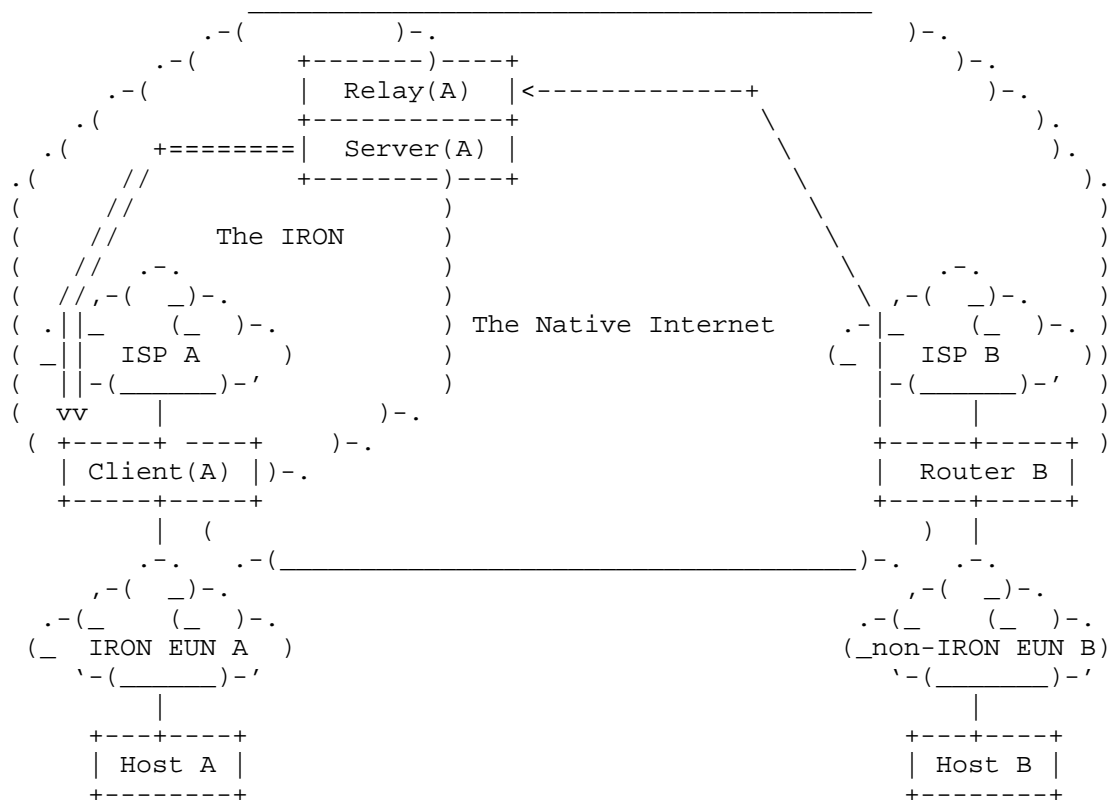


Figure 9: From Non-IRON Host B to IRON Host A

In this scenario, host B sends packets destined to host A via its network interface connected to non-IRON EUN B. Routing will direct the packets to Relay(A) which then forwards them to Server(A) using encapsulation if necessary.

Server(A) will then check its FIB to discover an entry that covers

destination address A with Client(A) as the next hop. Server(A) then (re-)encapsulates the packets in an outer header that uses the source address, destination address and nonce parameters associated with the tunnel to Client(A). Server(A) next releases these (re-)encapsulated packets into the Internet, where routing will direct them to Client(A). Client(A) will in turn decapsulate the packets and forward the inner packets to host A via its network interface connected to IRON EUN A.

This scenario always involves a Server and Relay owned by the VPC that provides service to IRON EUN A. It therefore imparts a cost that would need to be borne by either the VPC or its customers.

6.5. Mobility, Multihoming and Traffic Engineering Considerations

While IRON Servers and Relays can be considered as fixed infrastructure, Clients may need to move between different network points of attachment, connect to multiple ISPs, or explicitly manage their traffic flows. The following sections discuss mobility, multihoming and traffic engineering considerations for IRON client routers.

6.5.1. Mobility Management

When a Client changes its network point of attachment (e.g., due to a mobility event), it configures one or more new locators. If the Client has not moved far away from its previous network point of attachment, it simply informs its Server of any locator additions or deletions. This operation is performance-sensitive, and should be conducted immediately to avoid packet loss.

If the Client has moved far away from its previous network point of attachment, however, it re-issues the anycast discovery procedure described in Section 6.1 to discover whether its candidate set of Servers has changed. If the Client's current Server is also included in the new list received from the VPC, this provides indication that the Client has not moved far enough to warrant changing to a new Server. Otherwise, the Client may wish to move to a new Server in order to maintain optimal routing. This operation is not performance-critical, and therefore can be conducted over a matter of seconds/minutes instead of milliseconds/microseconds.

To move to a new Server, the Client first engages in the EP registration process with the new Server and maintains the registrations through periodic SRS/SRA exchanges the same as described in Section 6.1. The Client then informs its former Server that it has moved by providing it with the locator address of the new Server. The Client then discontinues the SRS/SRA keepalive process

with the former Server, which will garbage-collect the stale FIB entries when their lifetime expires. This will allow the former Server to redirect existing correspondents to the new Server so that no packets are lost.

Note that IRON addresses only network mobility and not host mobility. Mobility considerations for hosts within IRON EUNs are out of scope.

6.5.2. Multihoming

A Client may register multiple locators with its Server. It can assign metrics with its registrations to inform the Server of preferred locators, and can select outgoing locators according to its local preferences. Multihoming is therefore naturally supported.

6.5.3. Inbound Traffic Engineering

A Client can dynamically adjust the priorities of its prefix registrations with its Server in order to influence inbound traffic flows. It can also change between Servers when multiple Servers are available, but should strive for stability in its Server selection in order to limit VPC network routing churn.

6.5.4. Outbound Traffic Engineering

A Client can select outgoing locators, e.g., based on current QoS considerations such as minimizing one-way delay or one-way delay variance.

6.6. Renumbering Considerations

As new link layer technologies and/or service models emerge, customers will be motivated to select their service providers through healthy competition between ISPs. If a customer's EUN addresses are tied to a specific ISP, however, the customer may be forced to undergo a painstaking EUN renumbering process if it wishes to change to a different ISP [RFC4192][RFC5887].

When a customer obtains EP prefixes from a VPC, it can change between ISPs seamlessly and without need to renumber. If the VPC itself applies unreasonable costing structures for use of the EPs, however, the customer may be compelled to seek a different VPC and would again be required to confront a renumbering scenario. The IRON approach to renumbering avoidance therefore depends on VPCs conducting ethical business practices and offering reasonable rates.

6.7. NAT Traversal Considerations

The Internet today consists of a global public IPv4 routing and addressing system with non-IRON EUNs that use either public or private IPv4 addressing. The latter class of EUNs connect to the public Internet via Network Address Translators (NATs). When a Client is located behind a NAT, it selects Servers using the same procedures as for Clients with public addresses, i.e., it will send SRS messages to Servers in order to get SRA messages in return. The only requirement is that the Client must configure its SEAL encapsulation to use a transport protocol that supports NAT traversal, namely UDP.

Since the Server maintains state about its Client customers, it can discover locator information for each Client by examining the UDP port number and IP address in the outer headers of SRS messages. When there is a NAT in the path, the UDP port number and IP address in the SRS message will correspond to state in the NAT box and might not correspond to the actual values assigned to the Client. The Server can then encapsulate packets destined to hosts in the Client's EUN within outer headers that use this IP address and UDP port number. The NAT box will receive the packets, translate the values in the outer headers, then forward the packets to the Client. In this sense, the Server's "locator" for the Client consists of the concatenation of the IP address and UDP port number.

IRON does not introduce any new issues to complications raised for NAT traversal or for applications embedding address referrals in their payload.

6.8. Nested EUN Considerations

Each Client configures a locator that may be taken from an ordinary non-EPA address assigned by an ISP or from an EPA address taken from an EP assigned to another Client. In that case, the Client is said to be "nested" within the EUN of another Client, and recursive nestings of multiple layers of encapsulations may be necessary.

For example, in the network scenario depicted in Figure 10 Client(A) configures a locator EPA(B) taken from the EP assigned to EUN(B). Client(B) in turn configures a locator EPA(C) taken from the EP assigned to EUN(C). Finally, Client(C) configures a locator ISP(D) taken from a non-EPA address delegated by an ordinary ISP(D). Using this example, the "nested-IRON" case must be examined in which a host A which configures the address EPA(A) within EUN(A) exchanges packets with host Z located elsewhere in the Internet.

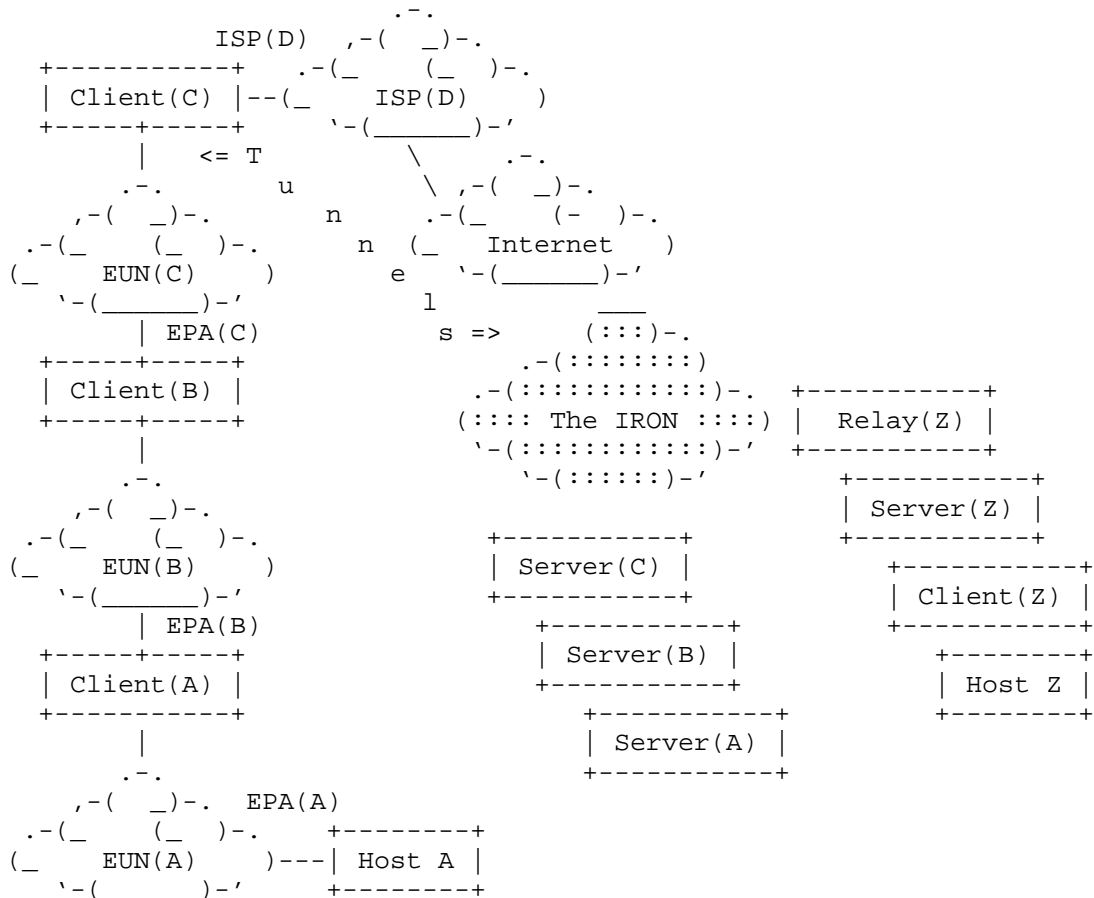


Figure 10: Nested EUN Example

The two cases of host A sending packets to host Z, and host Z sending packets to host A, must be considered separately as described below.

6.8.1. Host A Sends Packets to Host Z

Host A first forwards a packet with source address EPA(A) and destination address Z into EUN(A). Routing within EUN(A) will direct the packet to Client(A), which encapsulates it in an outer header with EPA(B) as the outer source address and Server(A) as the outer destination address then forwards the once-encapsulated packet into EUN(B). Routing within EUN[B] will direct the packet to Client(B), which encapsulates it in an outer header with EPA(C) as the outer source address and Server(B) as the outer destination address then forwards the twice-encapsulated packet into EUN(C). Routing within

EUN(C) will direct the packet to Client(C), which encapsulates it in an outer header with ISP(D) as the outer source address and Server(C) as the outer destination address. Client(C) then sends this triple-encapsulated packet into the ISP(D) network, where it will be routed into the Internet to Server(C).

When Server(C) receives the triple-encapsulated packet, it removes the outer layer of encapsulation and forwards the resulting twice-encapsulated packet into the Internet to Server(B). Next, Server(B) removes the outer layer of encapsulation and forwards the resulting once-encapsulated packet into the Internet to Server(A). Next, Server(A) checks the address type of the inner address 'Z'. If Z is a non-EPA address, Server(A) simply decapsulates the packet and forwards it into the Internet. Otherwise, Server(A) rewrites the outer source and destination addresses of the once-encapsulated packet and forwards it to Relay(Z). Relay(Z) in turn rewrites the outer destination address of the packet to the locator for Server(Z), then forwards the packet and sends a redirect to Server(A) (which forwards the redirect to Client(A)). Server(Z) then re-encapsulates the packet and forwards it to Client(Z), which decapsulates it and forwards the inner packet to host Z. Subsequent packets from Client(A) will then use Server(Z) as the next hop toward host Z, which eliminates Server(A) and Relay(Z) from the path.

6.8.2. Host Z Sends Packets to Host A

Whether or not host Z configures an EPA address, its packets destined to Host A will eventually reach Server(A). Server(A) will have a mapping that lists Client(A) as the next hop toward EPA(A). Server(A) will then encapsulate the packet with EPA(B) as the outer destination address and forward the packet into the Internet. Internet routing will convey this once-encapsulated packet to Server(B) which will have a mapping that lists Client(B) as the next hop toward EPA(B). Server(B) will then encapsulate the packet with EPA(C) as the outer destination address and forward the packet into the Internet. Internet routing will then convey this twice-encapsulated packet to Server(C) which will have a mapping that lists Client(C) as the next hop toward EPA(C). Server(C) will then encapsulate the packet with ISP(D) as the outer destination address and forward the packet into the Internet. Internet routing will then convey this triple-encapsulated packet to Client(C).

When the triple-encapsulated packet arrives at Client(C), it strips the outer layer of encapsulation and forwards the twice-encapsulated packet to EPA(C) which is the locator address of Client(B). When Client(B) receives the twice-encapsulated packet, it strips the outer layer of encapsulation and forwards the once-encapsulated packet to EPA(B) which is the locator address of Client(A). When Client(A)

receives the once-encapsulated packet, it strips the outer layer of encapsulation and forwards the unencapsulated packet to EPA(A) which is the host address of host A.

7. Additional Considerations

Considerations for the scalability of Internet Routing due to multihoming, traffic engineering and provider-independent addressing are discussed in [I-D.narten-radir-problem-statement]. Other scaling considerations specific to IRON are discussed in Appendix B.

Route optimization considerations for mobile networks are found in [RFC5522].

8. Related Initiatives

IRON builds upon the concepts RANGER architecture [RFC5720], and therefore inherits the same set of related initiatives.

Virtual Aggregation (VA) [I-D.ietf-grow-va] and Aggregation in Increasing Scopes (AIS) [I-D.zhang-evolution] provide the basis for the Virtual Prefix concepts.

Internet vastly improved plumbing (Ivip) [I-D.whittle-ivip-arch] has contributed valuable insights, including the use of real-time mapping. The use of Servers as mobility anchor points is directly influenced by Ivip's associated TTR mobility extensions [TTRMOB].

[I-D.bernardos-mext-nemo-ro-cr] discussed a route optimization approach using a Correspondent Router (CR) model. The IRON Server construct is similar to the CR concept described in this work, however the manner in which customer EUNs coordinates with Servers is different and based on the redirection model associated with NBMA links.

Numerous publications have proposed NAT traversal techniques. The NAT traversal techniques adapted for IRON were inspired by the Simple Address Mapping for Premises Legacy Equipment (SAMPLE) proposal [I-D.carpenter-software-sample].

9. IANA Considerations

There are no IANA considerations for this document.

10. Security Considerations

Security considerations that apply to tunneling in general are discussed in [I-D.ietf-v6ops-tunnel-security-concerns]. Additional considerations that apply also to IRON are discussed in RANGER [RFC5720], VET [I-D.templin-intarea-vet] and SEAL [I-D.templin-intarea-seal].

The IRON system further depends on mutual authentication of IRON Clients to Servers and Servers to Relays. This is accomplished through initial authentication exchanges followed by per-packet nonces that can be used to detect off-path attacks. As for all Internet communications, the IRON system also depends on Relays acting with integrity and not injecting false advertisements into the BGP (e.g., to mount traffic siphoning attacks).

Each VPC overlay network requires a means for assuring the integrity of the interior routing system so that all Relays and Servers in the overlay have a consistent view of Client<->Server bindings. Finally, DOS attacks on IRON Relays and Servers can occur when packets with spoofed source addresses arrive at high data rates. This issue is no different than for any border router in the public Internet today, however.

11. Acknowledgements

This ideas behind this work have benefited greatly from discussions with colleagues; some of which appear on the RRG and other IRTF/IETF mailing lists. Robin Whittle and Steve Russert co-authored the TTR mobility architecture which strongly influenced IRON. Eric Fleischman pointed out the opportunity to leverage anycast for discovering topologically-close Servers. Thomas Henderson recommended a quantitative analysis of scaling properties.

The following individuals provided essential review input: Mohamed Boucadair, John Buford, Wesley Eddy, Dae Young Kim and Robin Whittle.

12. References

12.1. Normative References

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

12.2. Informative References

- [BGPMON] net, B., "BGPmon.net - Monitoring Your Prefixes", <http://bgpmon.net/stat.php>", June 2010.
- [I-D.bernardos-mext-nemo-ro-cr]
Bernardos, C., Calderon, M., and I. Soto, "Correspondent Router based Route Optimisation for NEMO (CRON)", draft-bernardos-mext-nemo-ro-cr-00 (work in progress), July 2008.
- [I-D.carpenter-software-sample]
Carpenter, B. and S. Jiang, "Legacy NAT Traversal for IPv6: Simple Address Mapping for Premises Legacy Equipment (SAMPLE)", draft-carpenter-software-sample-00 (work in progress), June 2010.
- [I-D.ietf-grow-va]
Francis, P., Xu, X., Ballani, H., Jen, D., Raszuk, R., and L. Zhang, "FIB Suppression with Virtual Aggregation", draft-ietf-grow-va-03 (work in progress), August 2010.
- [I-D.ietf-v6ops-tunnel-security-concerns]
Hoagland, J., Krishnan, S., and D. Thaler, "Security Concerns With IP Tunneling", draft-ietf-v6ops-tunnel-security-concerns-02 (work in progress), August 2010.
- [I-D.narten-radir-problem-statement]
Narten, T., "On the Scalability of Internet Routing", draft-narten-radir-problem-statement-05 (work in progress), February 2010.
- [I-D.russert-rangers]
Russert, S., Fleischman, E., and F. Templin, "RANGER Scenarios", draft-russert-rangers-05 (work in progress), July 2010.
- [I-D.templin-intarea-seal]
Templin, F., "The Subnetwork Encapsulation and Adaptation Layer (SEAL)", draft-templin-intarea-seal-20 (work in progress), September 2010.
- [I-D.templin-intarea-vet]
Templin, F., "Virtual Enterprise Traversal (VET)", draft-templin-intarea-vet-16 (work in progress), July 2010.

- [I-D.whittle-ivip-arch]
Whittle, R., "Ivip (Internet Vastly Improved Plumbing) Architecture", draft-whittle-ivip-arch-04 (work in progress), March 2010.
- [I-D.zhang-evolution]
Zhang, B. and L. Zhang, "Evolution Towards Global Routing Scalability", draft-zhang-evolution-02 (work in progress), October 2009.
- [RFC1070] Hagens, R., Hall, N., and M. Rose, "Use of the Internet as a subnetwork for experimentation with the OSI network layer", RFC 1070, February 1989.
- [RFC2526] Johnson, D. and S. Deering, "Reserved IPv6 Subnet Anycast Addresses", RFC 2526, March 1999.
- [RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.
- [RFC3849] Huston, G., Lord, A., and P. Smith, "IPv6 Address Prefix Reserved for Documentation", RFC 3849, July 2004.
- [RFC4192] Baker, F., Lear, E., and R. Droms, "Procedures for Renumbering an IPv6 Network without a Flag Day", RFC 4192, September 2005.
- [RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [RFC4548] Gray, E., Rutenmiller, J., and G. Swallow, "Internet Code Point (ICP) Assignments for NSAP Addresses", RFC 4548, May 2006.
- [RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.
- [RFC5522] Eddy, W., Ivancic, W., and T. Davis, "Network Mobility Route Optimization Requirements for Operational Use in Aeronautics and Space Exploration Mobile Networks", RFC 5522, October 2009.
- [RFC5720] Templin, F., "Routing and Addressing in Networks with Global Enterprise Recursion (RANGER)", RFC 5720, February 2010.
- [RFC5737] Arkko, J., Cotton, M., and L. Vegoda, "IPv4 Address Blocks

Reserved for Documentation", RFC 5737, January 2010.

- [RFC5743] Falk, A., "Definition of an Internet Research Task Force (IRTF) Document Stream", RFC 5743, December 2009.
- [RFC5887] Carpenter, B., Atkinson, R., and H. Flinck, "Renumbering Still Needs Work", RFC 5887, May 2010.
- [TTRMOB] Whittle, R. and S. Russert, "TTR Mobility Extensions for Core-Edge Separation Solutions to the Internet's Routing Scaling Problem, <http://www.firstpr.com.au/ip/ivip/TTR-Mobility.pdf>", August 2008.

Appendix A. IRON VPs Over Internetworks with Different Address Families

The IRON architecture leverages the routing system by providing generally shortest-path routing for packets with EPA addresses from VPs that match the address family of the underlying Internetwork. When the VPs are of an address family that is not routable within the underlying Internetwork, however, (e.g., when OSI/NSAP [RFC4548] VPs are used within an IPv4 Internetwork) a global mapping database is required to allow Servers to map VPs to companion prefixes taken from address families that are routable within the Internetwork. For example, an IPv6 VP (e.g., 2001:DB8::/32) could be paired with a companion IPv4 prefix (e.g., 192.0.2.0/24) so that encapsulated IPv6 packets can be forwarded over IPv4-only Internetworks.

Every VP in the IRON must therefore be represented in a globally distributed Master VP database (MVPd) that maintains VP-to-companion prefix mappings for all VPs in the IRON. The MVPd is maintained by a globally-managed assigned numbers authority in the same manner as the Internet Assigned Numbers Authority (IANA) currently maintains the master list of all top-level IPv4 and IPv6 delegations. The database can be replicated across multiple servers for load balancing much in the same way that FTP mirror sites are used to manage software distributions.

Upon startup, each Server discovers the full set of VPs for the IRON by reading the MVPd. The Server reads the MVPd from a nearby server and periodically checks the server for deltas since the database was last read. After reading the MVPd, the Server has a full list of VP to companion prefix mappings.

The Server can then forward packets toward EPAs covered by a VP by encapsulating them in an outer header of the VP's companion prefix address family and using any address taken from the companion prefix

as the outer destination address. The companion prefix therefore serves as an anycast prefix.

Possible encapsulations in this model include IPv6-in-IPv4, IPv4-in-IPv6, OSI/CLNP-in-IPv6, OSI/CLNP-in-IPv4, etc.

Appendix B. Scaling Considerations

Scaling aspects of the IRON architecture have strong implications for its applicability in practical deployments. Scaling must be considered along multiple vectors including Interdomain core routing scaling, scaling to accommodate large numbers of customer EUNs, traffic scaling, state requirements, etc.

In terms of routing scaling, each VPC will advertise one or more VPs from which EPs are delegated to customer EUNs. Routing scaling will therefore be minimized when each VP covers many EPs. For example, the IPv6 prefix 2001:DB8::/32 contains 2^{24} ::/56 EP prefixes for assignment to EUNs. The IRON could therefore accommodate 2^{32} ::/56 EPs with only 2^8 ::/32 VPs advertised in the interdomain routing core.

In terms of traffic scaling for Relays, each Relay represents an ASBR of a "shell" enterprise network that simply directs arriving traffic packets with EPA destination addresses towards Servers that service customer EUNs. Moreover, the Relay sheds traffic destined to EPAs through redirection which removes it from the path for the vast majority of traffic packets. On the other hand, each Relay must handle all traffic packets forwarded between its customer EUNs and the non-IRON Internet. The scaling concerns for this latter class of traffic are no different than for ASBR routers that connect large enterprise networks to the Internet. In terms of traffic scaling for Servers, each Server services a set of the VPC overlay network's customer EUNs. The Server services all traffic packets destined to its EUNs but only services the initial packets of flows initiated from the EUNs and destined to EPAs. Therefore, traffic scaling for EPA-addressed traffic is an asymmetric consideration and is proportional to the number of EUNs each Server serves.

In terms of state requirements for Relays, each Relay maintains a list of all Servers in the VPC overlay network as well as FIB entries for all customer EUNs that each Server serves. This state is therefore dominated by the number of EUNs in the VPC overlay network. Sizing the Relay to accommodate state information for all EUNs is therefore required during VPC overlay network planning. In terms of state requirements for Servers, each Server maintains tunnel state for each of the customer EUNs it serves but need not keep state for

all EUNs in the VPC overlay network. Finally, neither Relays nor Servers need keep state for final destinations of outbound traffic.

Clients source and sink all traffic packets originating from or destined to the customer EUN. Therefore traffic scaling considerations for Clients are the same as for any site border router. Clients also retain state for the Servers for final destinations of outbound traffic flows. This can be managed as soft state, since stale entries purged from the cache will be refreshed when new traffic packets are sent.

Author's Address

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707 MC 7L-49
Seattle, WA 98124
USA

Email: fltemplin@acm.org

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 6, 2012

T. Tsou
Huawei Technologies (USA)
C. Zhou
T. Taylor
Huawei Technologies
Q. Chen
China Telecom
March 5, 2012

"Gateway-Initiated" 6rd
draft-tsou-softwire-gwinit-6rd-06

Abstract

This document proposes an alternative 6rd deployment model to that of RFC 5969. The basic 6rd model allows IPv6 hosts to gain access to IPv6 networks across an IPv4 access network using 6-in-4 tunnels. 6rd requires support by a device (the 6rd-CE) on the customer site, which must also be assigned an IPv4 address. The alternative model described in this document initiates the 6-in-4 tunnels from an operator-owned gateway collocated with the operator's IPv4 network edge, rather than from customer equipment. The advantages of this approach are that it requires no modification to customer equipment and avoids assignment of IPv4 addresses to customer equipment. The latter point means less pressure on IPv4 addresses in a high-growth environment.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 6, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	3
1.1. Requirements Language	3
2. Problem Statement	3
3. Proposed Solution	5
3.1. Prefix Delegation	6
3.2. Relevant Differences From Basic 6rd	7
4. IANA Considerations	7
5. Security Considerations	7
6. Acknowledgements	8
7. References	8
7.1. Normative References	8
7.2. Informative References	8
Authors' Addresses	8

1. Introduction

6rd ([RFC5969]) provides a transition tool for connecting IPv6 devices across an IPv4 network to an IPv6 network, at which point the packets can be routed natively. The network topology is shown in Figure 1.

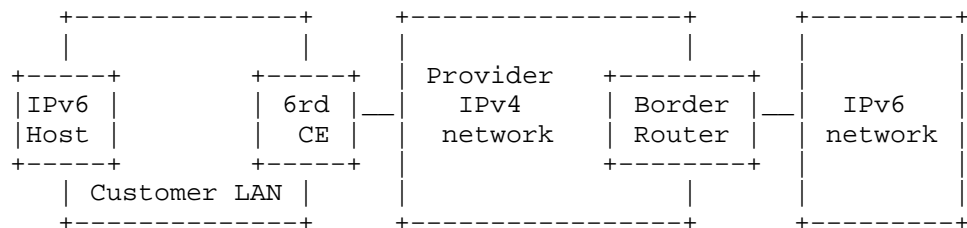


Figure 1: 6rd Deployment Topology

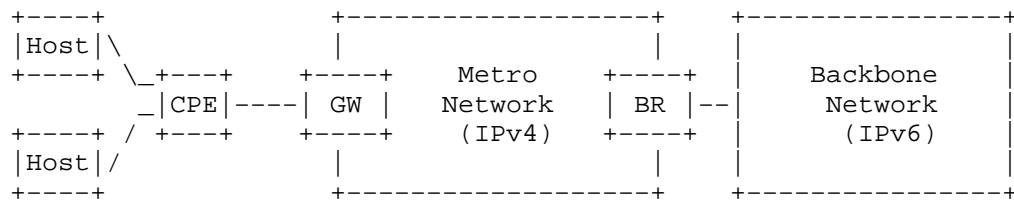
In Figure 1, the CE is the customer edge router. It is provisioned with a delegated IPv6 prefix, but also with an IPv4 address so that it is reachable through the IPv4 network. If a public IPv4 address is provisioned to every customer, it will aggravate the pressure due to IPv4 address shortage for operators faced with a high rate of growth in the number of broadband subscribers to their network. The use of private addresses with 6rd avoids this particular difficulty, but brings other complications.

1.1. Requirements Language

This document uses no requirements language.

2. Problem Statement

Consider an operator facing a high subscriber growth rate. As a result of this growth rate, the operator faces pressure on its stock of available public IPv4 addresses. For this reason, the operator is motivated to offer IPv6 access as quickly as possible. Figure 2 shows the sort of network situation envisioned in the present document.



Host = IPv6 customer host device
 CPE = customer edge device (customer-provided)
 GW = provider edge device (Gateway)
 BR = border router (dual stack)

Specialized GW and BR functions are described in the next section.

Figure 2: Typical Network Scenario For IPv6 Transition

The backbone network will be the first part of the operator's network to support IPv6. The metro network is not so easily upgraded to support IPv6 since many devices need to be modified and there may be some impact to existing services. Thus any means of providing IPv6 access has to minimize the changes required to devices in the metro network.

In contrast to the situation described for basic 6rd [RFC5569], the operator is assumed to have no control over the capabilities of the IP devices on the customer premises. As a result, the operator cannot assume that any of these devices are capable of supporting 6rd.

If the customer equipment is in bridged mode and IPv6 is deployed to sites via a Service Provider's (SP's) IPv4 network, the IPv6-only host needs a IPv6 address to visit the IPv6 service. In this scenario, 6to4 or 6rd can be used. However, each IPv6-only host may need one corresponding IPv4 address when using public IPv4 address in 6to4 or 6rd, which puts great address pressure on the operators.

If the CPE in the above figure is acting in bridging mode, each host behind it needs to be directly assigned an IPv6 prefix so it can access IPv6 services. If the CPE is acting in routing mode, only the CPE needs to be assigned an IPv6 prefix, and it delegates prefixes to the hosts behind it.

If the Gateway supports IPv4 only, then an IPv4 address must also be assigned to each host (bridging mode) or to the CPE (routing mode). Both cases, but bridging mode in particular, put pressure on the provider's stock of IPv4 addresses.

If the Gateway is dual stack, an arrangement may be possible whereby all communication between the Gateway and the customer site uses IPv6 and the need to assign IPv4 addresses to customer devices is avoided. A possible solution is presented in the next section.

3. Proposed Solution

For basic 6rd [RFC5969], the 6rd CE initiates the 6-in-4 tunnel to the 6rd Border Relay to carry its IPv6 traffic. To avoid the requirement for customer premises equipment to fulfill this role, it is necessary to move the tunneling function to a network device. This document identifies a functional element termed the 6rd Gateway to perform this task. In what follows, the 6rd Gateway and 6rd Border Relay are referred to simply as the Gateway and Border Relay respectively.

The functions of Gateway are:

- o to generate and allocate Gateway initiated 6rd delegated prefixes for IPv6-capable customer devices, as described in Section 3.1.
- o to forward outgoing IPv6 packets through a tunnel to a Border Relay, which extracts and forwards them to an IPv6 network as for 6rd;
- o to extract incoming IPv6 packets tunneled from the Border Relay and forward them to the correct user device.

In the proposed solution, there is only one tunnel initiated from each Gateway to the Border Relay, which greatly reduces the number of tunnels the Border Relay has to handle. The deployment scenario consistent with the problem statement in Section 2 collocates the Gateway with the IP edge of the access network. This is shown in Figure 2 above, and is the typical placement of the Broadband Network Gateway (BNG) in a fixed broadband network. By assumption, the metro network beyond the BNG is IPv4. Transport between the customer site and the Gateway is over layer 2.

The elements of the proposed solution are these:

- o The IPv6 prefix assigned to the customer site contains the compressed IPv4 address of the network-facing side of the Gateway, plus a manually provisioned or Gateway-generated customer site identifier. This is illustrated in Figure 3 below.
- o The Border Relay is able to route incoming IPv6 packets to the correct Gateway by extracting the compressed Gateway address from

the IPv6 destination address of the incoming packet, expanding it to a full 32-bit IPv4 address, and setting it as the destination address of the encapsulated packet.

- o The Gateway can route incoming packets to the correct link after decapsulation using a mapping from either the full IPv6 prefix or the customer site identifier extracted from that prefix to the appropriate link.

3.1. Prefix Delegation

Referring back to Figure 2, prefix assignment to the customer equipment occurs in the normal fashion through the Gateway/IP edge, using either DHCPv6 or SLAAC. Figure 3 illustrates the structure of the assigned prefix, and how the components are derived, within the context of a complete address.

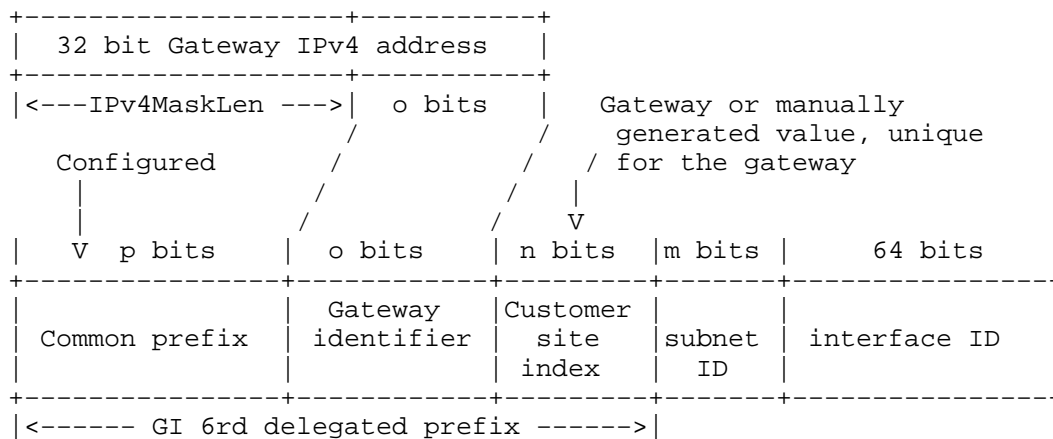


Figure 3: Gateway-Initiated 6rd Address Format for a Customer Site

The common prefix, i.e., the first p bits of the GI 6rd delegated prefix, is configured in the Gateway. This part of the prefix is common across multiple customers and multiple Gateways. Multiple common prefix values may be used in a network either for service separation or for scalability.

The Gateway Identifier is equal to the o low-order bits of the Gateway IPv4 address on the virtual link to the Border Relay. The number of bits o is equal to 32 - IPv4MaskLen, where the latter is the length of the IPv4 prefix from which the Gateway IPv4 addresses are derived. The value of IPv4MaskLen is configured in both the Gateways and the Border Relays.

The Customer Site Index is effectively a sequence number assigned to an individual customer site served by the Gateway. The value of the index for a given customer site must be unique across the Gateway. The length n of the Customer Site Index is provisioned in the Gateway, and must be large enough to accommodate the number of customer sites that the Gateway is expected to serve.

To give a numerical example, consider a 6rd domain containing ten million IPv6-capable customer devices (a rather high number given that 6rd is meant for the early stages of IPv6 deployment). The estimated number of 6rd Gateways needed to serve this domain would be in the order of 3,300, each serving 30,000 customer devices. Assuming best-case compression for the Gateway addresses, the Gateway Identifier field has length $o = 12$ bits. If IPv6-in-IPv4 tunneling is being used, this best case is more likely to be achievable than it would be if the IPv4 addresses belonged to the customer devices. More controllably, the customer device index has length $n = 15$ bits.

Overall, these figures suggest that the length p of the common prefix can be 29 bits for a /56 delegated prefix, or 21 bits if /48 delegated prefixes need to be allocated.

3.2. Relevant Differences From Basic 6rd

A number of the points in [RFC5969] apply with the simple substitution of the Gateway for the 6rd CE. When it comes to configuration, the definition of IPv4MaskLen changes, and there are other differences as indicated in the previous section. Since special configuration of customer equipment is not required, the 6rd DHCPv6 option is inapplicable.

Since the link for the customer site to the network now extends only as far as the Gateway, Neighbour Unreachability Detection on the part of customer devices is similarly limited in scope.

4. IANA Considerations

This memo includes no request to IANA.

5. Security Considerations

No change from [RFC5969].

6. Acknowledgements

Thanks to Ole Troan for his technical comments on an early version of this document.

7. References

7.1. Normative References

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

7.2. Informative References

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

Authors' Addresses

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

Phone:
Email: Tina.Tsou.Zouting@huawei.com

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

Phone:
Email: cathy.zhou@huawei.com

Tom Taylor
Huawei Technologies
Ottawa, Ontario
Canada

Phone:
Email: tom.taylor.stds@gmail.com

Qi Chen
China Telecom
109, Zhongshan Ave. West,
Tianhe District, Guangzhou 510630
P.R. China

Phone:
Email: chenqi.0819@gmail.com

