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

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".

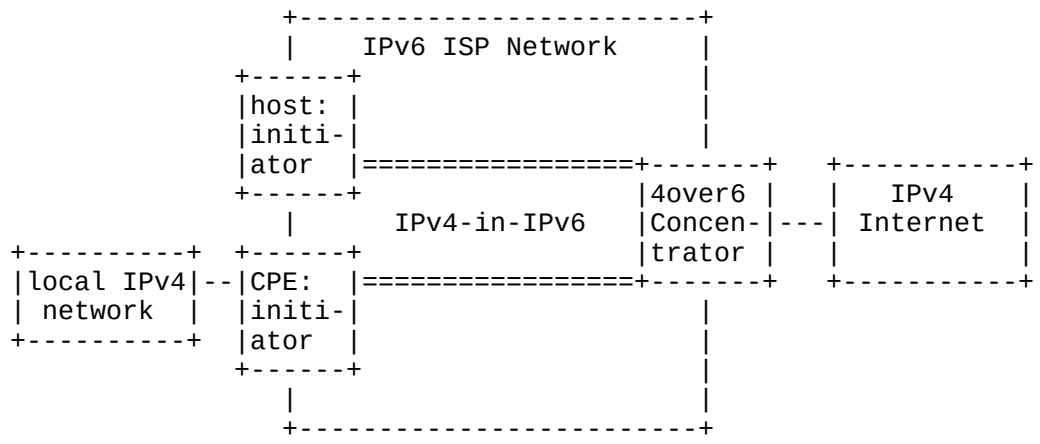


Figure 1 Public 4over6 scenario

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 softwire 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 44CGN 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-softwire-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 softwire 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-softwire-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: yiu_lee@cable.comcast.com

Softwire
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2012

Y. Cui
J. Dong
P. Wu
M. Xu
Tsinghua University
March 12, 2012

Softwire Mesh Management Information Base(MIB)
draft-cui-softwire-mesh-mib-04

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular it defines objects for managing softwire mesh [RFC5565].

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 13, 2012.

Copyright Notice

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

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

described in the Simplified BSD License.

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

Softwire mesh framework RFC 5565 [RFC5565]is a tunneling mechanism which enables connectivity between islands of IPv4, IPv6 or dual-stack networks across single IPv4 or IPv6 backbone networks. In softwire mesh solution, extended multiprotocol-BGP (MP-BGP)is used to set up tunnels and advertise prefixes among address family border routers (AFBRs).

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular it defines objects for managing softwire mesh [RFC5565].

2. The Internet-Standard Management Framework

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

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

3. Terminology

This document uses terminology from softwire problem statement RFC 4925 [RFC4925] and softwire mesh framework RFC5565 [RFC5565].

4. Conventions

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

5. Structure of the MIB Module

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

5.1. The swmSupportedTunnlTable Subtree

Since AFBR need to negotiate with BGP peer what kind of tunnel they will use, it should firstly announce the types of tunnels it

supports. The swmSupportedTunnlTable subtree provides the information. According to section 4 of RFC 5512[RFC5512], current softwire mesh tunnel types include IP-IP, GRE and L2TPv3.

5.2. The swmEncapsTable Subtree

The swmEncapsTable subtree provides softwire mesh NLRI-NH information about the AFBR. It indicates which I-IP destination address will be encapsulated according to the arriving packet's E-IP destination address. The definitions of E-IP and I-IP are explained in section 4.1 of RFC 5565[RFC5565].

5.3. The swmBGPNeighborTable Subtree

The subtree provides softwire mesh BGP neighbor information about the AFBR. It includes the address of softwire mesh BGP peer, and the kind of tunnel that the AFBR would use to communicate with this BGP peer.

5.4. The swmMIBConformance Subtree

The subtree provides conformance information of MIB objects.

6. Relationship to Other MIB Modules

6.1. Relationship to the IF-MIB

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

The ifOperStatus in ifTable would be used to represent whether the mesh function of the AFBR has been started. During the BGP OPEN phase, if the softwire mesh capability is negotiated, the mesh function could be considered to be started, and ifOperStatus is "up". Otherwise the ifOperStatus is "down".

If it is IPv4-over-IPv6 softwire mesh tunnel, the ifInUcastPkts will represent the number of IPv6 packets which can be decapsulated to IPv4 in the virtual interface. The ifOutUcastPkts contains the number of IPv6 packets which have been encapsulated with IPv4 packets in it. Particularly, if these IPv4 packets need to be fragmented, the number counted here is the packets after fragmentation.

If it is IPv6-over-IPv4 softwire mesh tunnel, the ifInUcastPkts stands for the number of IPv4 packets which would be decapsulated to IPv6 in the virtual interface. The ifOutUcastPkts represents the number of IPv4 packets which have been encapsulated from IPv6. Particularly, if these IPv6 packets need to be fragmented, the number counted here is the packets after fragmentation. Similar definition apply to other counting objects in ifTalbe.

6.2. Relationship to the IP Tunnel MIB

The IP Tunnel MIB [RFC4087] contains objects common to all IP tunnels, including softwire mesh. Additionally, tunnel encapsulation specific MIB (as is defined in this document) extends the IP tunnel MIB to further described encapsulation specific information.

Since softwire mesh is a point to multi-point tunnel, we need to specify an encapsulation table to support E-IP routing among AFBRs. With the encapsulation information, the correct forwarding of E-IP packets will be performed among AFBRs by using I-IP encapsulation. Each AFBR also needs to know information about remote BGP peers (AFBRs), so that these AFBRs can negotiate E-IP information and the tunnel types they support.

The implementation of the IP Tunnel MIB is required for softwire mesh. The tunnelIfEncapsMethod in the tunnelIfEntry should be set to softwireMesh("xx"), and corresponding entry in the softwire mesh MIB module will exist for every tunnelIfEntry with this tunnelIfEncapsMethod. The tunnelIfRemoteInetAddress must be set to 0.0.0.0 for IPv4 or :: for IPv6 because it is a point to multi-point tunnel.

Since tunnelIfAddressType in tunnelIfTable represents the type of address in the corresponding tunnelIfLocalInetAddress and tunnelIfRemoteInetAddress objects, we can also use the tunnelIfAddressType to specify the softwire mesh tunnel is IPv4-over-IPv6 or IPv6-over-IPv4. When tunnelIfAddressType is IPv4, the encapsulation would be IPv6- over-IPv4; When tunnelIfAddressType is IPv6, the encapsulation would be IPv4-over-IPv6.

6.3. MIB modules required for IMPORTS

The following MIB module IMPORTS objects from SNMPv2-SMI [RFC2578], SNMPv2-TC [RFC2579], SNMPv2-CONF [RFC2580], IF-MIB [RFC2863] and INET-ADDRESS-MIB [RFC4001].

7. Definitions

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

```
           IMPORTS
              TruthValue, TEXTUAL-CONVENTION
              TimeStamp
                  FROM SNMPv2-TC

              OBJECT-GROUP, MODULE-COMPLIANCE
                  FROM SNMPv2-CONF

              MODULE-IDENTITY, OBJECT-TYPE, mib-2, Unsigned32, Counter32,
              Counter64
                  FROM SNMPv2-SMI

              IANAtunnelType          FROM IANAifType-MIB;

              InetAddress, InetAddressPrefixLength
                  FROM INET-ADDRESS-MIB

           swmMIB MODULE-IDENTITY
               LAST-UPDATED "201112290000Z"        -- December 29, 2011
               ORGANIZATION "Softwire Working Group"
               CONTACT-INFO "

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

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

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

                    Mingwei Xu
                    Email:  xmw@cernet.edu.cn

                    Email comments directly to the softwire WG Mailing
                    List at softwires@ietf.org
                "

              DESCRIPTION
                  "This MIB module contains managed object definitions for
                   the softwire mesh framework."

              REVISION    "201203120000Z"
              DESCRIPTION
                  "draft-04 version"
          ::= {transmission xxx} --xxx to be replaced with correct value

              -- swmSupportedTunnelTable
```

```
        swmSupportedTunnelTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF swmSupportedTunnelEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                "A table of objects that shows what kind of tunnels
                can be supported in the AFBR."
            ::= { swmMIB 1 }

        swmSupportedTunnelEntry  OBJECT-TYPE
            SYNTAX      swmSupportedTunnelEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                "A set of objects that shows what kind of tunnels
                can be supported in the AFBR. If the AFBR supports
                several kinds of tunnel type, the
                swmSupportedTunnelTalbe would has several entries."
            INDEX { swmSupportedTunnelType }
            ::= { swmSupportedTunnelTable 1 }

        swmSupportedTunnelEntry ::=
            SEQUENCE {
                swmSupportedTunnelType              IANATunnelType
            }

        swmSupportedTunnelType OBJECT-TYPE
            SYNTAX      IANATunnelType
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Represents the kind of tunneling type that the AFBR
                support. "
            ::= { swmSupportedTunnelTypeEntry 1 }
        -- end of swmSupportedTunnelTable

        --swmEncapsTable
        swmEncapsTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF swmEncapsEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                "A table of objects that display and control the
                softwire mesh encapsulation information."
            ::= { swmMIB 2 }

        swmEncapsEntry  OBJECT-TYPE
            SYNTAX      swmEncapsEntry
```

```
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "A set of objects that display and control the
            softwire mesh encapsulation information."
        INDEX { ifIndex,
                swmEncapsEIPDst,
                swmEncapsEIPMask
              }
        ::= { swmEncapsTable 1 }

    swmEncapsEntry ::=
        SEQUENCE {
            swmEncapsEIPDst          InetAddress,
            swmEncapsEIPMask         InetAddressPrefixLength,
            swmEncapsIIPDst          InetAddress
        }

    swmEncapsEIPDst OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The destination E-IP address that decide which
            I-IP address will be encapsulated. The address Type
            is opposite to tunnelIfAddressType in tunnelIfTable."
        ::= { swmEncapsEntry 1 }

    swmEncapsEIPMask OBJECT-TYPE
        SYNTAX      InetAddressPrefixLength
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The prefix length of E-IP address."
        ::= { swmEncapsEntry 2 }

    swmEncapsIIPDst OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The I-IP address that will be encapsulated
             according to the E-IP address.The address Type
            is the same as tunnelIfAddressType in tunnelIfTable.
            Since the tunnelIfRemoteInetAddress in tunnelIfTable
            should be 0.0.0.0 or ::, swmEncapIIPDst is the
            destination address used in the outer IP header."
        ::= { swmEncapsEntry 3 }
```

```
        -- End of swmEncapsTable

        -- swmBGPNeighborTable
        swmBGPNeighborTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF swmBGPNeighborEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                "A table of objects that display the softwire mesh
                BGP neighbor information."
            ::= { swmMIB 3 }

        swmBGPNeighborEntry  OBJECT-TYPE
            SYNTAX      swmBGPNeighborEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                "A set of objects that display the softwire mesh
                BGP neighbor information."
            INDEX {
                    ifIndex,
                    swmBGPNeighborInetAddress
                  }
            ::= { swmBGPNeighborTable 1 }

        swmBGPNeighborEntry ::=
            SEQUENCE {
                swmBGPNeighborInetAddress        InetAddress,
                swmBGPNeighborTunnelType         IANATunnelType
            }

        swmBGPNeighborInetAddress OBJECT-TYPE
            SYNTAX      InetAddress
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "The address of the ABFR's BGP neighbor. The
                address type is the same as tunnelIfAddressType
                in tunnelIfTable"
            ::= { swmBGPNeighborEntry 1 }

        swmBGPNeighborTunnelType OBJECT-TYPE
            SYNTAX      IANATunnelType
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Represents the kind of tunneling type that the
                AFBR used to comunication with the BGP neighbor"
```

```
            ::= { swmBGPNeighborEntry 2 }
        -- End of swmBGPNeighborTable

    -- conformance information
     swmMIBConformance
                        OBJECT IDENTIFIER ::= { swmMIB 4 }
     swmMIBCompliances
                        OBJECT IDENTIFIER ::= { swmMIBConformance 1 }
     swmMIBGroups
                        OBJECT IDENTIFIER ::= { swmMIBConformance 2 }

    -- compliance statements
    swmMIBCompliance MODULE-COMPLIANCE
      STATUS current
      DESCRIPTION
          "Describes the requirements for conformance to the softwire
          mesh MIB."

      MODULE -- this module
      MANDATORY-GROUPS    {
                            swmSupportedTunnelGroup,
                            swmEncapsGroup,
                            swmBGPNeighborGroup
                          }
      ::= { swmMIBCompliances 1 }

    swmSupportedTunnelGroup    OBJECT-GROUP
      OBJECTS {
          swmSupportedTunnelType
      }
      STATUS  current
      DESCRIPTION
          "The collection of objects which are used to show
          what kind of tunnel the AFBR supports."
      ::= { swmMIBGroups 1 }

    swmEncapsGroup    OBJECT-GROUP
      OBJECTS {
                  swmEncapsEIPDst,
                  swmEncapsEIPMask,
                  swmEncapsIIPDst
      }
      STATUS  current
      DESCRIPTION
          "The collection of objects which are used to display
          softwire mesh encapsulation information."
      ::= { swmMIBGroups 2 }
```

```
    swmBGPNeighborGroup    OBJECT-GROUP
      OBJECTS {
                  swmBGPNeighborInetAddress,
                  swmBGPNeighborTunnelType
      }
      STATUS  current
      DESCRIPTION
          "The collection of objects which are used to display
           softwire mesh BGP neighbor information."
      ::= { swmMIBGroups 3 }

          END
```

8. Security Considerations

The swmMIB module can be used for configuration of certain objects, and anything that can be configured can be incorrectly configured, with potentially disastrous results. Because this MIB module reuses the IP tunnel MIB, the security considerations of the IP tunnel MIB is also applicable to the Softwire mesh MIB.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator's responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

9. IANA Considerations

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

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


         IANAtunnelType ::= TEXTUAL-CONVENTION
             SYNTAX     INTEGER {

                        softwireMesh ("XX")        -- softwire Mesh tunnel

                        }
```

10. References

10.1. Normative References

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

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

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

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

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

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

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

10.2. Informative References

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

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

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

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

10.3. URL References

[idguidelines] IETF Internet Drafts editor, "http://www.ietf.org/ietf/1id-guidelines.txt".

[idnits] IETF Internet Drafts editor, "http://www.ietf.org/ID-Checklist.html".

[xml2rfc] XML2RFC tools and documentation, "http://xml.resource.org".

[ops] the IETF OPS Area, "http://www.ops.ietf.org".

[ietf] IETF Tools Team, "http://tools.ietf.org".

Authors' Addresses

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

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

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

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

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

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

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

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

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: July 28, 2013

Dayong Guo
Sheng Jiang (Editor)
Huawei Technologies Co., Ltd
R. Despres
RD-IPtech
R. Maglione
Telecom Italia
January 24, 2013

RADIUS Attribute for 6rd

draft-ietf-softwire-6rd-radius-attrib-11.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 July 28, 2013.

Copyright Notice

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

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

Abstract

IPv6 Rapid Deployment (6rd) provides both IPv4 and IPv6 connectivity services simultaneously during the IPv4/IPv6 co-existence period. The Dynamic Host Configuration Protocol (DHCP) 6rd option has been defined to configure the 6rd Customer Edge (CE). However, in many networks, the configuration information may be stored in Authentication Authorization and Accounting (AAA) servers while user configuration is mainly acquired from a Broadband Network Gateway (BNG) through the DHCP protocol. This document defines a Remote Authentication Dial In User Service (RADIUS) attribute that carries 6rd configuration information from the AAA server to BNGs.

Table of Contents

1. Introduction

Recently providers have started to deploy IPv6 and to consider transition to IPv6. IPv6 Rapid Deployment (6rd) [RFC5969] provides both IPv4 and IPv6 connectivity services simultaneously during the IPv4/IPv6 co-existence period. 6rd is used to provide IPv6 connectivity service through legacy IPv4-only infrastructure. 6rd uses Dynamic Host Configuration Protocol (DHCP) [RFC2131] and the 6rd Customer Edge (CE) uses the DHCP 6rd option [RFC5969] to discover a 6rd border relay and to configure IPv6 prefix and address.

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

Since the 6rd configuration information is stored in AAA servers and user configuration is mainly through DHCP between BNGs and hosts/CEs, 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].

The terms 6rd Customer Edge (6rd CE) and 6rd Border Relay (BR) are defined in [RFC5969].

3. IPv6 6rd Configuration with RADIUS

Figure 1 illustrates how the RADIUS protocol and DHCP cooperate to provide 6rd CE with 6rd configuration information.

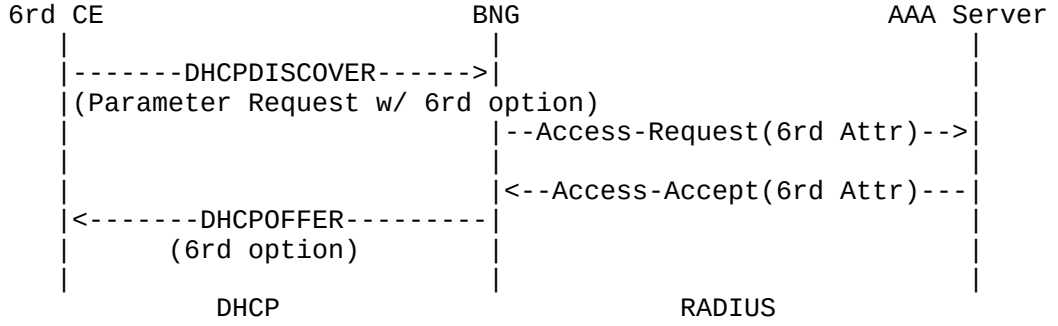


Figure 1: the cooperation between DHCP and RADIUS combining with RADIUS authentication

The BNG acts as a client of RADIUS and as a DHCP server. First, the 6rd CE MAY initiate a DHCPDISCOVER message that includes a Parameter Request option (55) [RFC2132] with the 6rd option [RFC5969]. When the BNG receives the DHCPDISCOVER, it SHOULD initiate a RADIUS Access-Request message, in which the User-Name attribute (1) SHOULD be filled by the 6rd CE MAC address, to the RADIUS server and the User-password (2) attribute SHOULD be filled by the shared 6rd password that has been preconfigured on the DHCP server, requesting authentication as defined in [RFC2865] with IPv6-6rd-Configuration attribute, defined in the next Section, in the desired attribute list. If the authentication request is approved by the AAA server, an Access-Accept message MUST be acknowledged with the IPv6-6rd-Configuration Attribute. Then, the BNG SHOULD respond to the 6rd CE with a DHCPOFFER message, which contains a DHCP 6rd option. The recommended format of the MAC address is as defined in Calling-Station-Id ([RFC3580] Section 3.20) without the SSID (Service Set Identifier) portion.

Figure 2 describes another scenario - later re-authorize - in which the authorization operation is not coupled with authentication. Authorization relevant to 6rd is done independently after the authentication process.

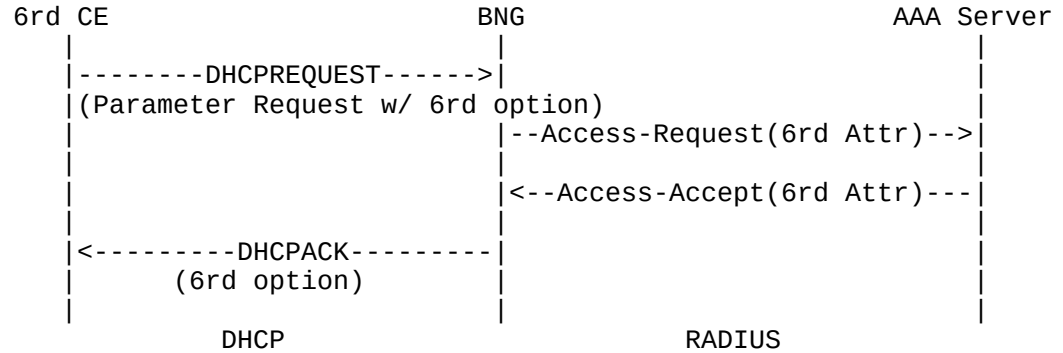


Figure 2: the cooperation between DHCP and RADIUS decoupled with RADIUS authentication

In this scenario, the Access-Request packet SHOULD contain a Service-Type attribute (6) with the value Authorize Only (17); thus, according to [RFC5080], the Access-Request packet MUST contain a State attribute that obtains from the previous authentication process.

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

After receiving the IPv6-6rd-Configuration Attribute in the initial Access-Accept, the BNG SHOULD store the received 6rd configuration parameters locally. When the 6rd CE sends a DHCP Request message to request an extension of the lifetime for the assigned address, the BNG does not have to initiate a new Access-Request towards the AAA server to request the 6rd configuration parameters. The BNG could retrieve the previously stored 6rd configuration parameters and use them in its reply.

If the BNG does not receive the IPv6-6rd-Configuration Attribute in the Access-Accept it MAY fall back to a pre-configured default 6rd configuration, if any. If the BNG does not have any pre-configured default 6rd configuration or if the BNG receives an Access-Reject, the tunnel cannot be established.

As specified in [RFC2131], section 4.4.5, "Reacquisition and expiration", if the DHCP server to which the DHCP Request message was sent at time T1 has not responded by time T2 (typically 0.375*duration_of_lease after T1), the 6rd CE (the DHCP client) SHOULD enter the REBINDING state and attempt to contact any server.

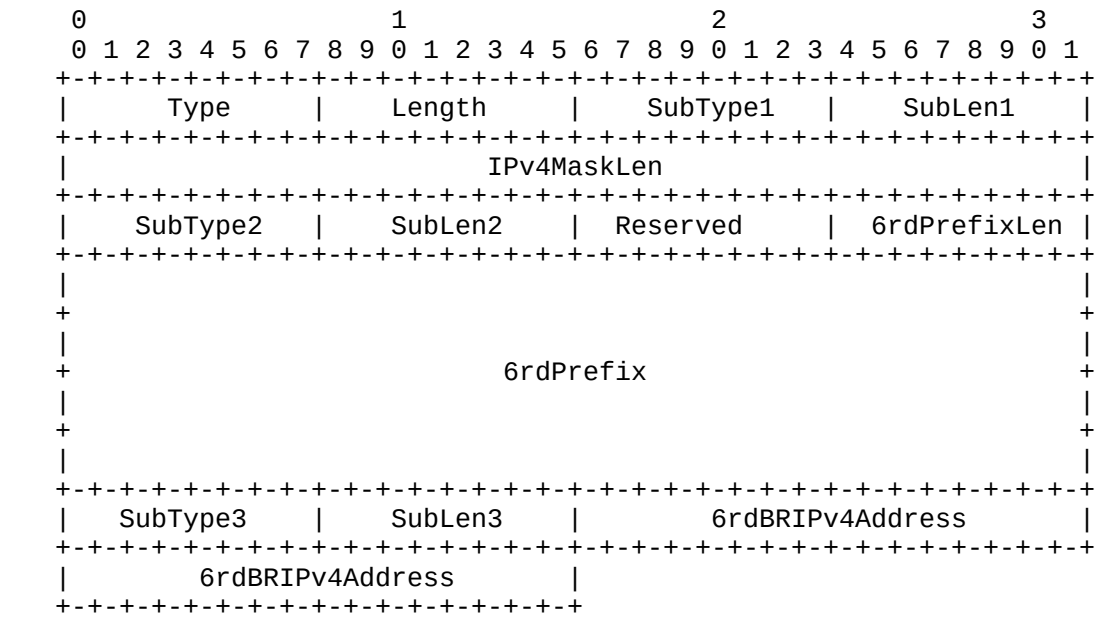
In this situation, the secondary BNG receiving the new DHCP message MUST initiate a new Access-Request towards the AAA server. The secondary BNG MAY include the IPv6-6rd-Configuration Attribute in its Access-Request.

4. Attributes

This section defines IPv6-6rd-Configuration Attribute which is used in the both abovementioned scenarios. The attribute design follows [RFC6158] and referring to [I-D.ietf-radext-radius-extensions].

4.1. IPv6-6rd-Configuration Attribute

The specification requires that multiple IPv4 addresses are associated with one IPv6 prefix. Given that RADIUS currently has no recommended way of grouping multiple attributes, the design below appears to be a reasonable compromise. The IPv6-6rd-Configuration Attribute is structured as follows:



Type

TBD

Length

28 + n*6 (the length of the entire attribute in octets; n stands for the number of BR IPv4 addresses, minimum n is 1).

SubType1

1 (SubType number, for the IPv4 Mask Length suboption)

SubLen1

6 (the length of the IPv4 Mask Length suboption)

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. Since [RFC6158] Section A.2.1 has forbidden 8-bit fields, a 32-bit field is used here.

SubType2

2 (SubType number, for the 6rd prefix suboption)

SubLen2

20 (the length of the 6rd prefix suboption)

Reserved

Set to be all 0 for now. Reserved for future use. To be compatible with other IPv6 prefix attributes in the RADIUS Protocol. The bits MUST be set to zero by the sender and MUST be ignored by the receiver.

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.

SubType3

3 (SubType number, for the 6rd Border Relay IPv4 address suboption)

SubLen3

6 (the length of the 6rd Border Relay IPv4 address suboption)

6rdBRIPv4Address

One or more IPv4 addresses of the 6rd Border Relay(s) for a given 6rd domain. The maximum RADIUS Attribute length of 255 octets results in a limit of 37 IPv4 addresses.

Since the subtypes have values, they can appear in any order. If multiple 6rdBRIPv4Address (subtype 3) appear, they are RECOMMENDED to be placed together.

The IPv6-6rd-Configuration Attribute is normally used in the Access-Accept messages. It MAY be used in Access-Request packets as a hint to the RADIUS server; for example if the BNG is pre-configured with a default 6rd configuration, these parameters MAY be inserted in the attribute. The RADIUS server MAY ignore the hint sent by the BNG and it MAY assign different 6rd parameters.

If the BNG includes the IPv6-6rd-Configuration Attribute, but the AAA server does not recognize it, this attribute MUST be ignored by the AAA Server.

If the BNG does not receive the IPv6-6rd-Configuration Attribute in the Access-Accept it MAY fallback to a pre-configured default 6rd configuration, if any. If the BNG does not have any pre-configured default 6rd configuration, the 6rd tunnel cannot be established.

If the BNG is pre-provisioned with a default 6rd configuration and the 6rd configuration received in Access-Accept is different from the configured default, then the 6rd configuration received in the Access-Accept message MUST be used for the session.

If the BNG cannot support the received 6rd configuration for any reason, the tunnel SHOULD NOT be established.

4.2. Table of attributes

The following table adds to the one in [RFC2865], Section 5.44, providing a guide to the quantity of IPv6-6rd-Configuration attributes that may be found in each kind of packet.

Request	Accept	Reject	Challenge	Accounting Request	#	Attribute
0-1	0-1	0	0	0-1	TBD	IPv6-6rd-Configuration
0-1	0-1	0	0	0-1	1	User-Name
0-1	0	0	0	0-1	2	User-Password
0-1	0-1	0	0	0-1	6	Service-Type
0-1	0-1	0-1	0-1	0-1	80	Message-Authenticator

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

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

5. Diameter Considerations

This attribute is usable within either RADIUS or Diameter [RFC6733]. Since the Attributes defined in this document will be allocated from the standard RADIUS type space, no special handling is required by Diameter entities.

6. Security Considerations

In 6rd scenarios, both CE and BNG are within a provider network, which can be considered as a closed network and a lower security threat environment. A similar consideration can be applied to the RADIUS message exchange between BNG and the AAA server.

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

A malicious user may use MAC address proofing and/or dictionary attack on the shared 6rd password that has been preconfigured on the DHCP server to get unauthorized 6rd configuration information. The follow-up secure issues have been considered in Section 12, [RFC5969].

Security considerations for 6rd specific between 6rd CE and BNG are discussed in [RFC5969]. Furthermore, generic DHCP security mechanisms can be applied DHCP intercommunication between 6rd CE and BNG.

Security considerations for the Diameter protocol are discussed in [RFC6733].

7. IANA Considerations

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

- IPv6-6rd-Configuration

IANA should allocate the number from the standard RADIUS Attributes range (values 1-191). The RFC Editor should use the assigned value to replace "TBD" in Sections 4.1 and 4.2, and should remove this paragraph.

8. Acknowledgments

The authors would like to thank Alan DeKok, Yong Cui, Leaf Yeh, Sean Turner, Joseph Salowey, Glen Zorn, Dave Nelson, Bernard Aboba, Benoit Claise, Barry Lieba, Stephen Farrell, Adrian Farrel, Ralph Droms and other members of Softwire WG, RADIUSExt WG, AAA-Doctors and Secdir for valuable comments.

9. References

9.1. Normative References

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

[RFC2131] R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, 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.

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

[RFC5080] Nelson, D. and DeKok A., "Common Remote Authentication Dial In User Service (RADIUS) Implementation Issues and Suggested Fixes", RFC 5080, December 2007.

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

[RFC6158] DeKok, A. and G. Weber, "RADIUS Design Guidelines", RFC 6158, March 2011.

[RFC6733] V. Fajardo, Ed., J. Arkko, J. Loughney, G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, October 2012.

9.2. Informative References

[RFC2607] Aboba, B. and J. Vollbrecht, "Proxy Chaining and Policy Implementation in Roaming", RFC 2607, June 1999.

[RFC2869] Rigney, C., Willats, W., and P. Calhoun, "RADIUS Extensions", RFC 2869, June 2000.

[RFC3580] Congdon, P., B. Aboba, A. Smith, G. Zorn and J. Roese, "IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines", RFC 3580, September 2003.

[I-D.ietf-radext-radius-extensions] DeKok, A. and A. Lior, "Remote Authentication Dial In User Service (RADIUS) Protocol Extensions", draft-ietf-radext-radius-extensions, work in process.

Author's Addresses

Dayong Guo
Huawei Technologies Co., Ltd
Q14 Huawei Campus, 156 BeiQi Road,
ZhongGuan Cun, Hai-Dian District, Beijing 100095
P.R. China
Email: guoseu@huawei.com

Sheng Jiang (Editor)
Huawei Technologies Co., Ltd
Q14 Huawei Campus, 156 BeiQi Road,
ZhongGuan Cun, Hai-Dian District, Beijing 100095
P.R. China
Email: jiangsheng@huawei.com

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

Roberta Maglione
Telecom Italia
Via Reiss Romoli 274
Torino 10148
Italy
Email: roberta.maglione@telecomitalia.it

Softwire
Internet-Draft
Intended status: Standards Track
Expires: September 8, 2011

D. Hankins
Google
T. Mrugalski
Gdansk University of Technology

March 7, 2011

Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite
draft-ietf-softwire-ds-lite-tunnel-option-10

Abstract

This document specifies a DHCPv6 option which is meant to be used by a Dual-Stack Lite Basic Bridging Broadband (B4) element to discover the IPv6 address of its corresponding Address Family Transition Router (AFTR).

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 8, 2011.

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. Requirements Language

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

2. Introduction

Dual-Stack Lite [I-D.softwire-ds-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, commonly referred to as a Softwire. A DS-Lite "Basic Bridging BroadBand" (B4) device will not know if the network it is attached to offers Dual-Stack Lite service, and if it did would not know the remote endpoint of the tunnel to establish a softwire.

To inform the B4 of the Address Family Transition Router’s (AFTR) location, a DNS [RFC1035] hostname 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.

To provide the conveyance of the configuration information, a single DHCPv6 [RFC3315] option is used, expressing the AFTR’s Fully Qualified Domain Name (FQDN) to the B4 element.

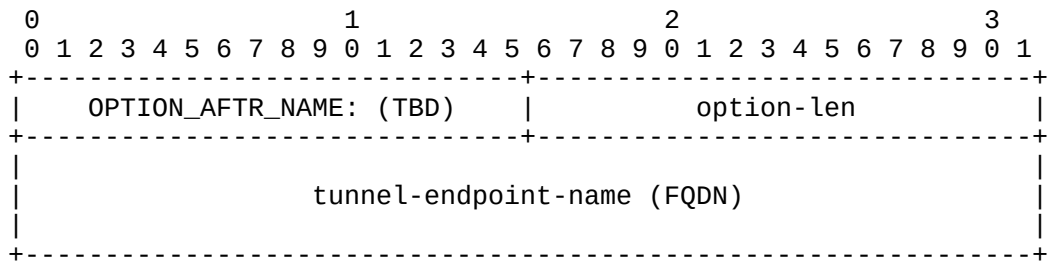
The details of how the B4 establishes an IPv4-in-IPv6 softwire to the AFTR are out of scope for this document.

3. The AFTR-Name DHCPv6 Option

The AFTR-Name option consists of option-code and option-len fields (as all DHCPv6 options have), and a variable length tunnel-endpoint-name field containing a fully qualified domain name that refers to the AFTR which the client MAY connect to.

The AFTR-Name option SHOULD NOT appear in any other than the following DHCPv6 messages: Solicit, Advertise, Request, Renew, Rebind, Information-Request and Reply.

The format of the AFTR-Name option is shown in the following figure:



OPTION_AFTR_NAME: (TBD)

option-len: Length of the tunnel-endpoint-name field in octets.

tunnel-endpoint-name: A fully qualified domain name of the AFTR tunnel endpoint.

Figure 1: AFTR-Name DHCPv6 Option Format

The tunnel-endpoint-name field is formatted as required in DHCPv6 [RFC3315] Section 8 ("Representation and Use of Domain Names"). Briefly, the format described is using a single octet noting the length of one DNS label (limited to at most 63 octets), followed by the label contents. This repeats until all labels in the FQDN are exhausted, including a terminating zero-length label. Any updates to Section 8 of DHCPv6 [RFC3315] also apply to encoding of this field. An example format for this option is shown in Figure 2, which conveys the FQDN "aftr.example.com.".

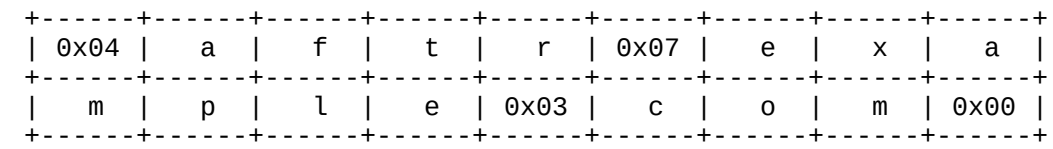


Figure 2: Example tunnel-endpoint-name.

Note that in the specific case of the example tunnel-endpoint-name (Figure 2), the length of the tunnel-endpoint-name is 18 octets, and so an option-len field value of 18 would be used.

The option is validated by confirming that all of the following conditions are met:

1. the option-len is greater than 3;

2. the option data can be contained by the option length (that the option length does not run off the end of the packet);

3. the individual label lengths do not exceed the option length;

4. the tunnel-endpoint-name is of valid format as described in DHCPv6 Section 8 [RFC3315];

5. there are no compression tags;

6. there is at least one label of nonzero length.

4. DHCPv6 Server Behavior

A DHCPv6 server SHOULD NOT send more than one AFTR-Name option. It SHOULD NOT permit the configuration of multiple names within one AFTR-Name option. Both of these conditions are handled exceptionally by the client, so an operator using software that does not perform these validations should be careful not to configure multiple domain names.

RFC 3315 Section 17.2.2 [RFC3315] describes how a DHCPv6 client and server negotiate configuration values using the Option Request Option (OPTION_ORO). As a convenience to the reader, we mention here that a server will not reply with a AFTR-Name option if the client has not explicitly enumerated it on its Option Request Option.

5. DHCPv6 Client Behavior

A client that supports the B4 functionality of DS-Lite (defined in [I-D.softwire-ds-lite]) and conforms to this specification MUST include OPTION_AFTR_NAME on its OPTION_ORO.

Because it requires DNS name to address resolution, the client MAY also wish to include the OPTION_DNS_SERVERS [RFC3646] option on its OPTION_ORO.

If the client receives the AFTR-Name option, it MUST verify the option contents as described in Section 3.

Note that in different environments, the B4 element and DHCPv6 client may be integrated, joined, or separated by a third pieces of software. For the purpose of this specification, we refer to the "B4 system" when specifying implementation steps that may be processed at any stage of integration between the DHCPv6 client software and the B4 element it is configuring.

If the B4 system receives more than one AFTR-Name option, it MUST use only the first instance of that option.

If the AFTR-Name option contains more than one FQDN, as distinguished by the presence of multiple root labels, the B4 system MUST use only the first FQDN listed in configuration.

The B4 system performs standard DNS resolution using the provided FQDN to resolve a AAAA Resource Record, as defined in [RFC3596] and STD 13 [RFC1034] [RFC1035].

If any DNS response contains more than one IPv6 address, the B4 system picks only one IPv6 address and uses it as a remote tunnel endpoint for the interface being configured in the current message exchange. The B4 system MUST NOT establish more than one DS-Lite tunnel at the same time per interface. For a redundancy and high availability discussion, see Section 12.3 "High availability" of [I-D.softwire-ds-lite].

Note that a B4 system may have multiple network interfaces, and these interfaces may be configured differently; some may be connected to networks that call for DS-Lite, and some may be connected to networks that are using normal dual stack or other means. The B4 system should approach this specification on an interface-by-interface basis. For example, if the B4 system is attached to multiple networks that provide the AFTR Name option, then the B4 system MUST configure a tunnel for each interface separately as each DS-Lite tunnel provides IPv4 connectivity for each distinct interface. Means to bind a AFTR Name and DS-Lite tunnel configuration to a given interface in a multiple interfaces device are out of scope of this document.

6. Security Considerations

This document does not present any new security issues, but as with all DHCPv6-derived configuration state, it is completely possible that the configuration is being delivered by a third party (Man In The Middle). As such, there is no basis to trust that the access the DS-Lite Softwire connection represents can be trusted, and it should not therefore bypass any security mechanisms such as IP firewalls.

RFC 3315 [RFC3315] discusses DHCPv6-related security issues.

[I-D.softwire-ds-lite] discusses DS-Lite related security issues.

7. IANA Considerations

IANA is requested to allocate single DHCPv6 option code referencing this document, delineating OPTION_AFTR_NAME.

8. Acknowledgements

Authors would like to thank Alain Durand, Rob Austein, Dave Thaler, Paul Selkirk, Ralph Droms, Mohamed Boucadair, Roberta Maglione and Shawn Routhier for their valuable feedback and suggestions.

This work has been partially supported by the Polish Ministry of Science and Higher Education under the European Regional Development Fund, Grant No. POIG.01.01.02-00-045/09-00 (Future Internet Engineering Project).

9. Normative References

[I-D.softwire-ds-lite] Durand, A., Ed., "Dual-stack lite broadband deployments post IPv4 exhaustion", draft-ietf-softwire-dual-stack-lite (work in progress).

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

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

[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.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.

[RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.

Authors' Addresses

David W. Hankins
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
USA

Email: dhankins@google.com

Tomasz Mrugalski
Gdansk University of Technology
ul. Storczykowa 22B/12
Gdansk 80-177
Poland

Phone: +48 698 088 272
Email: tomasz.mrugalski@eti.pg.gda.pl

softwire
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2012

R. Maglione
Telecom Italia
A. Durand
Juniper Networks

October 17, 2011

RADIUS Extensions for Dual-Stack Lite
draft-ietf-softwire-dslite-radius-ext-07

Abstract

Dual-Stack Lite is a solution to offer both IPv4 and IPv6 connectivity to customers which are addressed only with an IPv6 prefix. Dual-Stack Lite requires to pre-configure the Dual-Stack Lite Address Family Transition Router (AFTR) tunnel information on the Basic Bridging BroadBand (B4) element. In many networks, the customer profile information may be stored in Authentication Authorization and Accounting (AAA) servers while client configurations are mainly provided through Dynamic Host Configuration Protocol (DHCP). This document specifies a new Remote Authentication Dial In User Service (RADIUS) attribute to carry Dual-Stack Lite Address Family Transition Router Tunnel name; the RADIUS attribute is defined based on the equivalent DHCPv6 OPTION_AFTR_NAME option. This RADIUS attribute is meant to be used between the RADIUS Server and the Network Access Server (NAS), it is not intended to be used directly between the Basic Bridging BroadBand 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 19, 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.

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

Dual-Stack Lite [RFC6333] 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 Dual-Stack-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 Basic Bridging BroadBand (B4) of the Address Family Transition Router's (AFTR) location, a 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.

[RFC6334] specifies a DHCPv6 option which is meant to be used by a Dual-Stack Lite client (Basic Bridging BroadBand element, B4) to discover its Address Family Transition Router (AFTR) name. In order to be able to populate such option the DHCPv6 Server must be pre-provisioned with the Address Family Transition Router (AFTR) name.

In Broadband environments, customer profile may be managed by AAA servers, together with user Authentication, Authorization, and Accounting (AAA). Remote Authentication Dial In User Service (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 Dynamic Host Configuration Protocol (DHCP) running between the NAS and the requesting clients, a new RADIUS attribute is needed to send AFTR information from AAA server to the NAS.

This document aims at defining a new RADIUS attribute to be used for carrying the DS-Lite Tunnel Name, based on the equivalent DHCPv6 option already specified in [RFC6334]

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 [RFC6333]

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 when a PPP Session is used to provide connectivity to the user.

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 Name. When the NAS receives a DHCPv6 client request containing the DS-Lite tunnel Option, the NAS SHALL use the name returned in the RADIUS DS-Lite-Tunnel-Name attribute to populate the DHCPv6 OPTION_AFTR_NAME option in the DHCPv6 reply message.

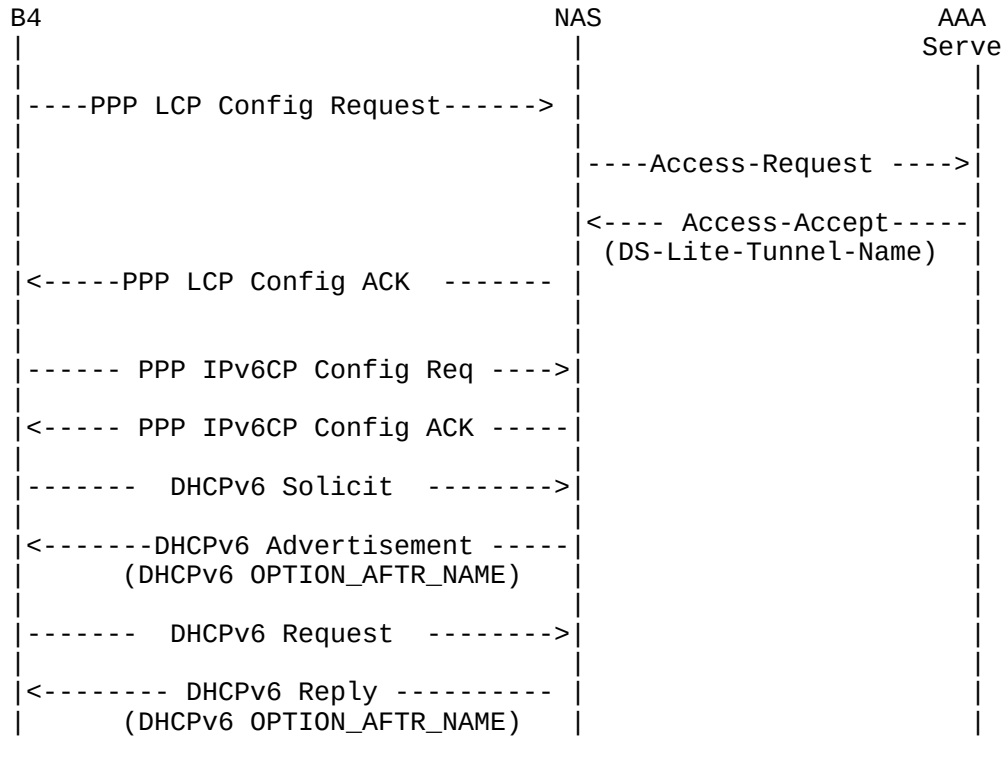


Figure 1: RADIUS and DHCPv6 Message Flow for a PPP Session

The Figure 2 illustrates how the RADIUS protocol and DHCPv6 work together to accomplish DS-Lite configuration on the B4 element when an IP Session is used to provide connectivity to the user.

The only difference between this message flow and previous one is that in this scenario the interaction between NAS and AAA/ RADIUS Server is triggered by the DHCPv6 Solicit message received by the NAS from the B4 acting as DHCPv6 client, while in case of a PPP Session the trigger is the PPP LCP Config Request message received by the NAS.

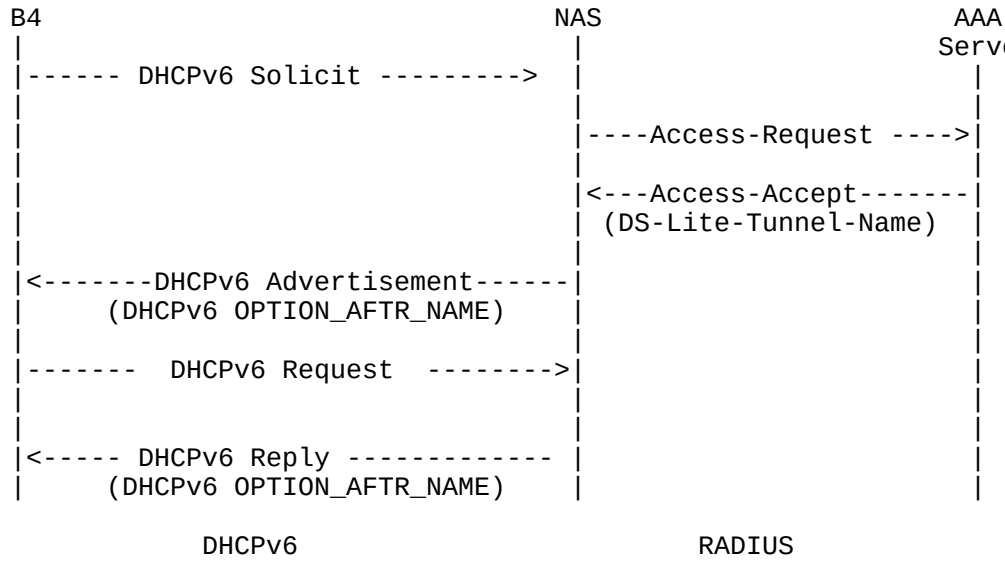


Figure 2: RADIUS and DHCPv6 Message Flow for an IP Session

In the scenario depicted in Figure 2 the Access-Request packet contains a Service-Type attribute with the value Authorize Only (17), thus according to [RFC5080] the Access-Request packet MUST contain a State attribute.

After receiving the DS-Lite-Tunnel-Name in the initial Access-Accept the NAS MUST store the received AFTR Tunnel Name locally. When the B4 sends a DHCPv6 Renew message to request an extension of the lifetimes for the assigned address or prefix, the NAS does not have to initiate a new Access-Request towards the AAA server to request the AFTR tunnel name. The NAS retrieves the previously stored AFTR tunnel name and uses it in its reply.

According to [RFC3315] if the DHCPv6 server to which the DHCPv6 Renew message was sent at time T1 has not responded, the DHCPv6 client initiates a Rebind/Reply message exchange with any available server. In this scenario the NAS receiving the DHCPv6 rebind message MUST initiate a new Access-Request towards the AAA server. The NAS MAY include the DS-Lite-Tunnel-Name attribute in its Access-Request.

If the NAS does not receive the DS-Lite-Tunnel-Name attribute in the Access-Accept it MAY fallback to a pre-configured default tunnel name, if any. If the NAS does not have any pre-configured default tunnel name or if the NAS receives an Access-Reject, the IPv4 over IPv6 tunnel cannot be established, thus the B4 element has only IPv6 connectivity.

4. RADIUS Attribute

This section specifies the format of the new RADIUS attribute.

4.1. 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_AFTR_NAME option [RFC6334]

This attribute MAY be used in Access-Request packets as a hint to the RADIUS server; for example if the NAS is pre-configured with a default tunnel name, this name MAY be inserted in the attribute. The RADIUS server MAY ignore the hint sent by the NAS and it MAY assign a different AFTR tunnel name.

If the NAS includes the DS-Lite-Tunnel-Name attribute, but the AAA server does not recognize it, this attribute MUST be ignored by the AAA Server.

If the NAS does not receive DS-Lite-Tunnel-Name attribute in the Access-Accept it MAY fallback to a pre-configured default tunnel name, if any. If the NAS does not have any pre-configured default tunnel name, the tunnel can not be established.

If the NAS is pre-provisioned with a default AFTR tunnel name and the AFTR tunnel name received in Access-Accept is different from the configured default, then the AFTR tunnel name received in the Access-Accept message MUST be used for the session.

If the NAS cannot support the received AFTR tunnel name for any reason, the tunnel SHOULD NOT be established.

When the Access-Request is triggered by a DHCPv6 Rebind message if the AFTR tunnel name received in the Access-Accept is different from the currently used one for that session, the NAS MUST force the B4 to re-establish the tunnel using the new AFTR name received in the Access-Accept message.

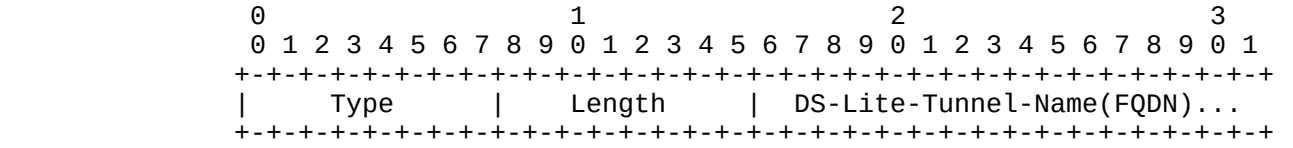
If an implementation includes the Change-of-Authorization (CoA) messages [RFC5176], they could be used to modify the current established DS-Lite tunnel. When the NAS receives a CoA Request message containing the DS-Lite-Tunnel-Name attribute, the NAS MUST send a Reconfigure message to a B4 to inform the B4 that the NAS has

new or updated configuration parameters and that the B4 is to initiate a Renew/Reply or Information-request/Reply transaction with the NAS in order to receive the updated information.

Upon receiving an AFTR tunnel name different from the currently used one, the B4 MUST terminate the current DS-Lite tunnel and the B4 MUST establish a new DS-LITE tunnel with the specified AFTR.

The DS-Lite-Tunnel-Name RADIUS attribute 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 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:

TBA1 for DS-Lite-Tunnel-Name.

Length:

This field indicates the total length in octets of this attribute including the Type, the Length fields and the length in octets of the DS-Lite-Tunnel-Name field

DS-Lite-Tunnel-Name:

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

As the DS-Lite-Tunnel-Name attribute is used to populate the DHCPv6 OPTION_AFTR_NAME option, the DS-Lite-Tunnel-Name field is formatted as required in DHCPv6 (Section 8 of [RFC3315] "Representation and Use of Domain Names"). Briefly, the format described is using a single octet noting the length of one DNS label (limited to at most 63 octets), followed by the label contents. This repeats until all labels in the FQDN are exhausted, including a terminating zero-length label. Any updates to Section 8 of [RFC3315] also apply to encoding of this field.

The data type of DS-Lite-Tunnel-Name RADIUS attribute is a string with opaque encapsulation, according to section 5 of [RFC2865]

5. Table of attributes

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

Access-Request	Access-Accept	Access-Reject	Challenge	Accounting-Request	#	Attribute
0-1	0-1	0	0	0-1	TBA1	DS-Lite-Tunnel-Name

CoA-Request	CoA-ACK	CoA-NACK	#	Attribute
0-1	0	0	TBA1	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] for RADIUS protocol and in [RFC5176] for CoA messages.

[RFC6333] discusses Dual-Stack Lite related security issues.

7. IANA Considerations

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

DS-Lite-Tunnel-Name - TBA1

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.

[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.

[RFC5080] Nelson, D. and A. DeKok, "Common Remote Authentication Dial In User Service (RADIUS) Implementation Issues and Suggested Fixes", RFC 5080, December 2007.

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

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

8.2. Informative 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-05 (work in progress), July 2011.

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

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:

SOFTWIRE WG
Internet-Draft
Intended status: Standards Track
Expires: October 30, 2012

F. Brockners
S. Gundavelli
Cisco
S. Speicher
Deutsche Telekom AG
D. Ward
Cisco

April 28, 2012

Gateway Initiated Dual-Stack Lite Deployment
draft-ietf-softwire-gateway-init-ds-lite-08

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 October 30, 2012.

Copyright Notice

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

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

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

Table of Contents

1. Overview

Gateway-Initiated Dual-Stack lite (GI-DS-lite) is a variant of the Dual-Stack lite (DS-lite) [RFC6333], 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 [RFC6275], 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 [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 softwire 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 Address Family Transition Router (AFTR) 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., GRE based encapsulation mechanisms is chosen, it allows the network between the access gateway and the AFTR 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. 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 [RFC4925]

SWID: Softwire Identifier

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 Appendix A.

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 SWID does not need to be globally unique, i.e. different SWIDs could be used to identify a softwire at the different ends of a softwire. 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, Section 6 for details. A Context-Identifier (CID) is used to multiplex flows associated with the individual access devices onto the softwire. Deployment dependent, the flows from a particular AD can be identified using either the source IP-address or an access tunnel identifier. 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 must be unique between gateway and AFTR to identify the flows associated with an AD. 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 softwire 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. Section 6 for details on different softwire types supported with GI-DS-lite.

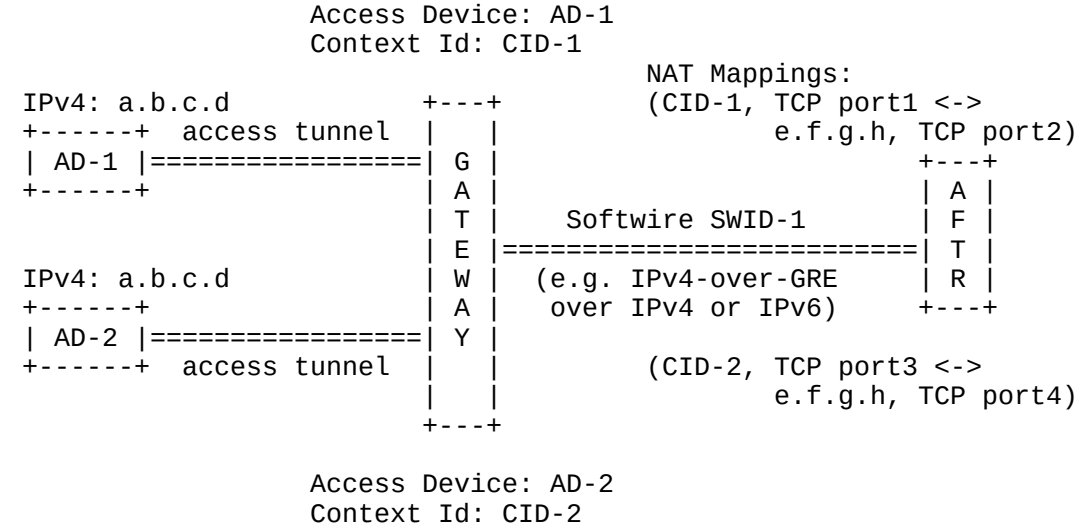


Figure 1: Gateway-initiated dual-stack lite reference architecture

The AFTR combines softwire termination and IPv4-IPv4 NAT. The NAT binding of the AD's address could be 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.

Softwire-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

- Depending on the embodiment of the CID (e.g. for GRE-encapsulation with GRE-key), 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, SHOULD be extended to include the CID and the identifier of the softwire (SWID).

- When creating an IPv4 to IPv4 NAT binding for an IPv4 packet flow received from the Gateway over the softwire, the AFTR SHOULD associate the CID with that NAT binding. It SHOULD use the combination of CID and SWID as the unique identifier and use those parameters in the NAT binding entry.

- When forwarding a packet to the access device, the AFTR SHOULD obtain the CID from the NAT binding associated with that flow. E.g., in case of GRE-encapsulation, it SHOULD add the CID to the GRE Key and Sequence number extension of the GRE header and tunnel it to the Gateway.

- On receiving any packet from the softwire, the AFTR SHOULD obtain the CID from the incoming packet and use it for performing the NAT binding look up and for performing the packet translation before forwarding the packet.

- The Gateway, on receiving any IPv4 packet from the access device SHOULD lookup the CID for that access device. In case of GRE encapsulation it can for example add the CID to the GRE Key and Sequence number extension of the GRE header and tunnel it to the

AFTR.

- On receiving any packet from the softwire, the Gateway SHOULD obtain the CID from the packet and use it for making the forwarding decision.

- When encapsulating an IPv4 packet, Gateway and AFTR SHOULD 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.

- The softwire between the Gateway and the AFTR MAY be created at system startup time, OR dynamically established on-demand. 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.

- 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.

- The softwire peers SHOULD use the correct path MTU value for the tunnel path between the access gateway and the AFTR. This value MAY be statically configured at softwire creation time, or dynamically discovered using the standard path MTU discovery techniques.

- 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 [RFC4364], or plain IP-in-IP encapsulation can be used. Softwire identification and Context-ID depend on the tunneling technology employed:

- GRE with GRE-key: SWID is the tunnel identifier of the GRE tunnel between the GW and the AFTR. The CID is the GRE-key associated with the AD.

- MPLS VPN: The SWID is a generic identifier which uniquely identifies the VPN at either the Gateway or AFTR. Depending on whether the Gateway or AFTR are acting as customer edge (CE) or, provider edge (PE), the SWID could e.g. be an attachment circuit identifier, an identifier representing the set of VPN route labels pointing to the routes within the VPN, etc. The AD's IPv4-address is the CID. For a given VPN, the AD's IPv4 address must be unique.

- IPv4/IPv6-in-MPLS: The SWID is the top MPLS label. CID might be the next MPLS label in the stack, if present, or the IP address of the AD.

- IPv4-in-IPv4: SWID is the outer IPv4 source address. The AD's IPv4 address is the CID. For a given outer IPv4 source address, the AD's IPv4 address must be unique.

- IPv4-in-IPv6: SWID is the outer IPv6 source address. If the AD's IPv4 address is used as CID, the AD's IPv4 address must be unique. If the IPv6-Flow-Label [RFC6437] is used as CID, the IPv4 addresses of the ADs may overlap. Given that the IPv6-Flow-Label is 20-bit wide, which is shorter than the recommended 32-bit CID, large scale deployments may require additional scaling considerations. In addition, one should ensure sufficient randomization of the IP-Flow-Label to avoid possible interference with other uses of the IP-Flow-Label, such as Equal Cost Multipath (ECMP) support.

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:

- 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.

  * "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 IPv4.

* "MPLS": Gateway and AFTR are connected by a MPLS network

Softwire	IPv4 address			Network-type			
	up	op	s	v4	v6	v4v6	MPLS
GRE with GRE-key	x	x	x	x	x	x	
MPLS VPN	x	x					x
IPv4/IPv6-in-MPLS	x	x	x				x
IPv4-in-IPv4	x			x			
IPv4-in-IPv6	x				x		
IPv4-in-IPv6 w/ FL	x	x	x		x		

Figure 3: Tunnel modes and their applicability

7. IANA Considerations

This specification does not require any IANA actions.

8. Security Considerations

The approach specified in this document allows the use of Dual-stack lite for tunnel-based access architectures. Rather than initiating the tunnel from the access device, GI-DS-lite logically extends the already existing access tunnel beyond the access gateway towards the Address Family Transition Router, and builds a virtual softwire between the AFTR and the access device. This approach requires the use of an additional context identifier in the AFTR and at the access gateway, which is required for making IP packet forwarding decisions.

A packet when received with an Incorrect context identifier at the access gateway/AFTR will result in associating the packet to an incorrect access device. Therefore, care must be taken to ensure an IP packet tunneled between the access gateway and the AFTR is carried

with the context identifier of the access device associated with that IP packet. The context identifier is not carried from the access device and it is not possible for one access device to claim the context identifier of some other access device. However, It is possible an on-path attacker between the access gateway and the AFTR can potentially modify the context identifier in the packet, resulting in association of the packet to an incorrect access device. This threat is no different from an on-path attacker modifying the source/destination address of an IP packet. However, this threat can be prevented by enabling IPsec security with integrity protection turned on, between the access gateway and the AFTR, that will ensure the correct binding of the context identifier and the inner packet. This specification does not require any other new security considerations other than those specified in dual-stack lite specification [RFC6333], and in the security considerations specified for the given access architecture, such as Proxy Mobile IPv6, leveraging this transitioning scheme.

9. Acknowledgements

The authors would like to acknowledge the discussions on this topic with Mark Grayson, Jay Iyer, Kent Leung, Vojislav Vucetic, Flemming Andreasen, 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, Cathy Zhou, Olaf Bonness, Paco Cortes, Jim Guichard, Stephen Farrell, Pete Resnik, Ralph Droms.

10. References

10.1. Normative References

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

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

[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.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private

Networks (VPNs)", RFC 4364, February 2006.

[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.

[RFC5555] Soliman, H., "Mobile IPv6 Support for Dual Stack Hosts and Routers", RFC 5555, June 2009.

[RFC5844] Wakikawa, R. and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6", RFC 5844, May 2010.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

[RFC6275] Perkins, C., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.

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

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, November 2011.

10.2. Informative References

[I-D.draft-ietf-dime-nat-control] Brockners, F., Bhandari, S., Singh, V., and V. Fajardo, "Diameter NAT Control Application", August 2009.

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

Appendix A. GI-DS-lite deployment

A.1. Connectivity establishment: Example call flow

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

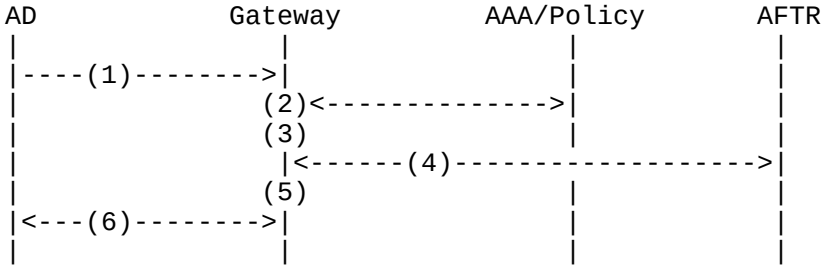


Figure 4: Example call flow for session establishment

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.

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 [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 that need to be tunneled towards the AFTR with the softwire 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).

A.2. GI-DS-lite applicability: Examples

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

- Mobile IP based access architectures: In a DSMIPv6 [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.

- Proxy Mobile IPv6 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 [RFC5844] functionality.

- 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.

- Fixed WiMAX architecture: If GI-DS-lite is applied to fixed WiMAX, the ASN-Gateway will implement the GI-DS-lite Gateway function.

- Mobile WiMAX: If GI-DS-lite is applied to mobile WiMAX, the home agent will implement the Gateway function.

- 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.

- In broadband access architectures using per-subscriber VLANs the BNG will implement the GI-DS-lite Gateway function.

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
Cisco
170 West Tasman Drive
SAN JOSE, CA 95134
USA

Email: wardd@cisco.com

Softwire
Internet-Draft
Intended status: Informational
Expires: January 12, 2012

Y. Lee
Comcast
R. Maglione
Telecom Italia
C. Williams
MCSR Labs
C. Jacquenet
M. Boucadair
France Telecom
July 11, 2011

Deployment Considerations for Dual-Stack Lite
draft-lee-softwire-dslite-deployment-02

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 considerations and applicability of the Dual-Stack Lite architecture.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 12, 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. 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

2.1. Interface Consideration

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. AFTR 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 recommended to configure two individual interfaces (i.e. one dedicated for IPv4 and one dedicated for IPv6) to segregate the functions.

2.2. MTU Considerations

DS-Lite is part tunneling protocol. Tunneling introduces some additional complexity and has a risk of MTU. With tunneling comes additional header overhead that implies that the tunnel's MTU is smaller than the raw interface MTU. The issue that the end user may experience is that they cannot download Internet pages or transfer files using File Transfer Protocol (FTP).

To mitigate the tunnel overhead, the access network could increase the MTU size to account the necessary tunnel overhead which is the size of an IPv6 header. If the access network MTU size is fixed and cannot be changed, the B4 element and the AFTR must support fragmentation.

2.3. Fragmentation

The IPv4-in-IPv6 tunnel is between B4 and AFTR. When a host behind the B4 element communicates to a server, both the host and the server are not aware of the tunnel. They may continue to use the maximum MTU size for communication. In fact, the IPv4 packet isn't over-sized, it is the v6 encapsulation that may cause the oversize. 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 oversized datagram. If the DF is set, the B4 element should send an ICMP "Destination Unreachable" with "Fragmentation Needed and Don't Fragment was Set" and drop the packet. If the DF is not set, the B4 element should fragment the IPv6 packet after the encapsulation. This mechanism is transport protocol agnostic and works for both UDP and TCP.

[editor note: Should we drop the IPv4 packet when DF is set?]

2.4. Lawful Intercept Considerations

Because of its IPv4-in-IPv6 tunneling scheme, interception of IPv4 sessions in DS-Lite architecture must be performed on the AFTR. Subjects can be uniquely identified by the IPv6 address assigned to the B4 element. Operator must associate the B4's IPv6 address and the public IPv4 address and port used by the subject.

Monitoring of a single subject may mean statically mapping the subject to a certain range of ports on a single IPv4 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 monitoring purposes to simplify such procedures. This requires to create a static mapping of a B4 element's IPv6 address to an IPv4 address that used for lawful intercept.

2.5. Logging at the AFTR

The timestamped logging is essential 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. Without time-specific logs of the address and port mappings, a misbehaving user stays well hidden behind the AFTR.

In DS-Lite framework, each B4 element is given a unique IPv6 address. The AFTR uses this IPv6 address to identify the B4 element. Thus, the AFTR must log the B4's IPv6 address and the IPv4 information. There are two types of logging: (1) Source-Specific Log and (2) Destination-Specific Log. For Source-Specific Log, the AFTR must timestamped log the B4's IPv6 address, transport protocol, source IPv4 address after NAT-ed, and source port. If a range of ports is dynamically assigned to a B4 element, the AFTR may create one log per range of ports to aggregate number of log entries. For Destination-Specific Log, the AFTR must timestamped log the B4's IPv6 address, transport protocol, source IPv4 address after NAT-ed, source port, destination address and destination port. The AFTR must log every

session from the B4 elements. No log aggregation can be performed. When using Destination-Specific Log, the operator must be careful of the large number of log entries created by the AFTR.

2.6. Blacklisting a shared IPv4 Address

AFTR is a NAT device. It shares a single IPv4 address with multiple users. [I-D.ietf-intarea-shared-addressing-issues] discusses many considerations when sharing address. When a pubic IPv4 address is blacklisted, this may affect multiple users and there is no effective way for the B4 element to notify the AFTR an IP address is blacklisted. It is recommended the server must no longer rely solely on IP address to identify an abused user. The server should combine the information stored in the transport layer (e.g. source port) and application layer (e.g. HTTP) to identify an abused user. [I-D.boucadair-intarea-nat-reveal-analysis] analyzes different approaches to identify a user in a shared address environment.

2.7. AFTR's Policies

There are two types of AFTR polices: (1) Outgoing Policies and (2) Incoming Policies. The outgoing policies must be implemented on the AFTR's internal interface connected to the B4 elements. The policies may include ACL and QoS settings. For example: the AFTR may only accept B4's connections originated from the IPv6 prefixes provisioned in the AFTR. The AFTR may also give priority to the packets marked by certain DSCP values. The AFTR may also limit the rate of port creation from a single B4's IPv6 address. Outgoing policies could be applied to individual B4 element or a set of B4 elements.

The incoming policies must be implemented on the AFTR's external interface connected to the IPv4 network. Similar to the outgoing policies, the policies may include ACL and QoS settings. Incoming policies are usually more general and globally applied to all users rather than individual user.

2.8. 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 ATFR 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 ATFR 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]

2.9. 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.

2.10. 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 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.

AFTR is a tunnel concentrator, B4 traffic must pass through the AFTR to reach the IPv4 Internet. Managing tunnels and NAT could be resource intensive, so the placement of the AFTR would affect the traffic flows in the access network and have operation implications. In general, there are two placements to deploy AFTR. Model One is to deploy the AFTR in the edge of network to cover a small region. Model Two is to deploy the AFTR in the core of network to cover a large region.

When the operator consider where to deploy the AFTR, they must make trade-offs. AFTR in Model One serves few B4 elements, thus, it requires less powerful AFTR. Moreover, the traffic flows are more evenly distributed to the AFTRs. However, it requires to deploy more AFTRs to cover the entire network. Often the operation cost increases proportionally to the number of network equipment. AFTR in Model Two covers larger area, thus, it serves more B4 elements. The operator could deploy only few AFTRs in the strategic locations to support the entire subscriber base. However, this model requires more powerful AFTR to sustain the load at peak hours. Since the AFTR would support B4 elements from different regions, the AFTR would be deployed deeper in the network and steer more traffic flows to the network where the AFTR is located.

DS-Lite framework can be incrementally deployed. An operator may consider to start with Model Two. When the demand increases, they could push the AFTR closer to the edge which would effectively become Model One.

2.11. AFTR Considerations for Geographically Aware Services

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 hundreds of machines, homes, and offices related only in that they are behind the same AFTR. Identification by IP address becomes more difficult and thus applications that assume such geographic information may not work as intended.

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 could 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. Another possibility the application could rely on location information such as GPS co-ordination to identify the user's location. This technique is commonly used in mobile deployment where the mobile devices are probably behind a NAT device.

2.12. 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 different types of traffic. DS-Lite tunnel can be seen as particular case of uniform conceptual tunnel model described in section 3.1 of [RFC2983]. The uniform model views an IP tunnel as just a necessary mechanism to get traffic to its destination, but the tunnel has no significant impact on traffic conditioning. In this model, any packet has exactly one DS Field that is used for traffic conditioning at any point and it is the field in the outermost IP header. In DS-Lite model this is the Traffic Class field in IPv6 header. According to [RFC2983] implementations of this model copy the DS value to the outer IP header at encapsulation and copy the outer header's DSCP value to the inner IP header at decapsulation. Applying the described model to DS-Lite scenario, it is recommended that the AFTR propagates the DSCP value in the IPv4 header to the IPv6 header after the encapsulation for the downstream traffic and, in the same way, the B4 propagates the DSCP value in the IPv4 header to the IPv6 header after the encapsulation for the upstream traffic.

2.13. 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.ietf-pcp-base] is designed to address this issues.

2.14. DS-Lite Tunnel Security

Section 11 of [I-D.ietf-softwire-dual-stack-lite] describes security issues associated to DS-Lite mechanism. One of the recommendations contained in this section, in order to limit service offered by AFTR only to registered customers, is to implement IPv6 ingress filter on

the AFTR's tunnel interface to accept only the IPv6 address range defined in the filter. This approach requires to know in advance the IPv6 prefix delegated to the customers in order to be able to configure the filter.

An alternative way to achieve the same goal and to provide some form of access control to the DS-Lite tunnel, is to use DHCPv6 Leasequery defined in [RFC5007]. When the AFTR receives a packet from an unknown (new) prefix it issues a DHCPv6 Leasequery based on IPv6 address to the DHCPv6 server in order to verify if that prefix was previously delegated by the DHCPv6 server to that specific client. The DHCPv6 Server will reply with the delegated prefix and the associated lease. If the two prefix are the same the ATFR accepts the packet otherwise it drops it and it denies the service.

2.15. IPv6-only Network considerations

In environments where the service provider wants to deploy AFTR in the IPv6 core network, the AFTR nodes may not have direct IPv4 connectivity. In this scenario the service provider extends the IPv6-only boundary to the border of the network and only the border routers have IPv4 connectivity. For both scalability and performance purposes AFTR capabilities are located in the IPv6-only core closer to B4 elements. The service provider assigns only IPv6 prefixes to the B4 capable devices but also continues to provide IPv4 services to these customers. In this scenario the AFTR has only IPv6-connectivity and must be able to send and receive IPv4 packets. Enhancements to the DS-LITE AFTR are required to achive this. [I-D.boucadair-softwire-dslite-v6only] describes such issues and enhancements to DS-Lite in IPv6-only deployments.

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/RADUIS 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 umber, 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-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-dslite-radius-ext]
Maglione, R. and A. Durand, "RADIUS Extensions for Dual-Stack Lite", draft-ietf-softwire-dslite-radius-ext-02 (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.

[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.

[RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng, "DHCPv6 Leasequery", RFC 5007, September 2007.

8.2. Informative References

[I-D.boucadair-intarea-nat-reveal-analysis]
Boucadair, M., Touch, J., Levis, P., and R. Penno, "Analysis of Solution Candidates to Reveal a Host Identifier in Shared Address Deployments", draft-boucadair-intarea-nat-reveal-analysis-03 (work in progress), June 2011.

[I-D.boucadair-softwire-dslite-v6only]
Boucadair, M., Jacquenet, C., Grimault, J., Kassi-Lahlou, M., Levis, P., Cheng, D., and Y. Lee, "Deploying Dual-Stack Lite in IPv6 Network", draft-boucadair-softwire-dslite-v6only-01 (work in progress), April 2011.

[I-D.ietf-intarea-server-logging-recommendations]
Durand, A., Gashinsky, I., Lee, D., and S. Sheppard, "Logging recommendations for Internet facing servers", draft-ietf-intarea-server-logging-recommendations-04 (work in progress), April 2011.

[I-D.ietf-intarea-shared-addressing-issues]
Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", draft-ietf-intarea-shared-addressing-issues-05 (work in progress), March 2011.

[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., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", draft-ietf-v6ops-ipv6-cpe-router-09 (work in progress), December 2010.

[I-D.xu-behave-stateful-nat-standby]
Xu, X., Boucadair, M., Lee, Y., and G. Chen, "Redundancy Requirements and Framework for Stateful Network Address Translators (NAT)", draft-xu-behave-stateful-nat-standby-06 (work in progress), October 2010.

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

[RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.

[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: yiu_lee@cable.comcast.com
URI: http://www.comcast.com

Roberta Maglione
Telecom Italia
Via Reiss Romoli 274
Torino 10148
Italy

Email: roberta.maglione@telecomitalia.it
URI:

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

Softwire WG
Internet-Draft
Intended status: Standards Track
Expires: December 15, 2011

Q. Wang
China Telecom
J. Qin
ZTE
M. Boucadair
C. Jacquenet
France Telecom
Y. Lee
Comcast
June 13, 2011

Multicast Extensions to DS-Lite Technique in Broadband Deployments
draft-qin-softwire-dslite-multicast-04

Abstract

This document proposes a solution for the delivery of multicast service offerings to DS-Lite serviced customers. The proposed solution relies upon a stateless IPv4-in-IPv6 encapsulation scheme and does not require performing any NAT operation along the path used to deliver multicast traffic.

Status of this Memo

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

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

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

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

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

DS-Lite [I-D.ietf-softwire-dual-stack-lite] is a technique to rationalize the use of the remaining IPv4 addresses during the transition period. The current design of DS-Lite covers unicast services exclusively.

If customers access IPv4 multicast-based service offerings through a DS-Lite environment, AFTR (Address Family Transition Router) devices have to process all the IGMP reports [RFC2236] [RFC3376] received within IPv4-in-IPv6 tunnels and behave as a replication point for downstream multicast traffic. That is likely to severely affect the multicast traffic forwarding efficiency by losing the benefits of deterministic replication of the data as close to the receivers as possible. As a consequence, the downstream bandwidth will be vastly consumed while the AFTR capability may become rapidly overloaded, in particular if the AFTR capability is deployed in a centralized manner.

This document discusses an extension to the DS-Lite model to be used for the delivery of IPv4 multicast-based service offerings.

1.1. Requirements Language

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

2. Terminology

This document makes use of the following terms:

- IPv4-embedded IPv6 address: is an IPv6 address which embeds a 32 bit-encoded IPv4 address. An IPv4-embedded IPv6 address can be unicast or multicast.

- mPrefix64: is a dedicated multicast IPv6 prefix for constructing IPv4-embedded IPv6 multicast address [I-D.boucadair-behave-64-multicast-address-format]. mPrefix64 can be of two types: ASM_mPrefix64 used in ASM mode or SSM_mPrefix64 used in SSM mode [RFC4607].

- uPrefix64: is a dedicated unicast IPv6 prefix for constructing IPv4-embedded IPv6 unicast address [RFC6052].

- Multicast AFTR (mAFTR for short): is a functional entity which is part of both the IPv4 and IPv6 multicast distribution trees and

which replicates IPv4 multicast streams into IPv4-in-IPv6 streams in the relevant branches of the IPv6 multicast distribution tree.

- Multicast B4 (mB4 for short): is a functional entity embedded in a CPE, which is able to enforce an IGMP-MLD interworking function (refer to Section 6.1) together with a de-capsulation function of received multicast IPv4-in-IPv6 packets.

3. Context and Scope

3.1. IPTV-centric View

IPTV generally includes two categories of service offerings:

1. VoD (Video on Demand) or Catch-up TV channels streams that are delivered using unicast mode to receivers.

2. Live TV Broadcast services that are generally multicast to receivers.

Numerous players intervene in the delivery of this service:

- Content Providers: the content can be provided by the same provider as the one providing the connectivity service or by distinct providers;

- Network Provider: the one providing network connectivity service (e.g., responsible for carrying multicast flows from head-ends to receivers). Refer to [I-D.ietf-mboned-multiaaa-framework].

Many of the current IPTV contents are likely to remain IPv4-formatted and out of control of the network providers. Additionally, there are numerous legacy receivers (e.g., IPv4-only Set Top Boxes (STB)) that can’t be upgraded or be easily replaced. As a consequence, IPv4 service continuity must be guaranteed during the transition period, including the delivery of multicast-based services such as Live TV Broadcasting. The dilemma is the same as in the transition of unicast-based Internet services where the customer premises and global Internet are out of control of the service providers even if they would like to promote the use of IPv6. The DS-Lite design tries to eliminate this issue by decoupling the IPv6 deployments in service provider networks from that in global Internet and in customer devices and applications.

DS-Lite can be seen as a catalyst for IPv6 deployment while preserving customer’s Quality of Experience (QoE). This is also the design goal of the solution proposed in this document for DS-Lite

serviced customers who have subscribed to a multicast-based service offering.

3.2. Scope

This document focuses only on issues raised by a DS-Lite networking environment: subscription to an IPv4 multicast group and the delivery of IPv4-formatted content to IPv4 receivers. In particular, only the following case is covered:

1. An IPv4 receiver accessing IPv4 content (i.e., content formatted and reachable in IPv4)

A viable scenario for this use case in DS-Lite environment: Customers with legacy receivers must continue to access the IPv4-enabled multicast services. This means the traffic should be accessed through IPv4 and additional functions are needed to traverse operators' IPv6- enabled network which is the purpose of this document. While since technically, there is no extra function required for the scenario of native access (i.e. to access dual-stack content natively from the IPv6 receiver), this portion is not taken into account. Refer to [I-D.jaclee-behave-v4v6-mcast-ps] for the deployment considerations.

This document does not cover the case where an IPv4 host connected to a CPE served by a DS-Lite AFTR can be the source of multicast traffic.

Note that some contract agreements prevent a network provider to alter the content as sent by the content provider, in particular for copyright, confidentiality and SLA assurance reasons. The streams should be delivered unaltered to requesting users.

4. Solution Overview

In the original DS-Lite specification [I-D.ietf-softwire-dual-stack-lite], an IPv4-in-IPv6 tunnel is used to carry the bidirectional IPv4 unicast traffic between B4 and AFTR. This document defines an IPv4-in-IPv6 encapsulation scheme to deliver multicast traffic. Within the context of this document, an IPv4 derived IPv6 multicast address is used as the destination of the encapsulated unidirectional IPv4-in-IPv6 multicast traffic from the mAFTR to the mB4. The IPv4 address of the source of the multicast content is represented in the IPv6 realm with an IPv4-embedded IPv6 address as well.

See following sections for the multicast distribution tree

establishment (Section 4.3) and the multicast traffic forwarding (Section 4.4).

Note that IPv4-in-IPv6 encapsulated multicast flows are treated in an IPv6 realm like any other IPv6 multicast flow. Upon completion of the establishment of a multicast distribution tree, no extra function is required to be defined to forward IPv4-in-IPv6 multicast traffic in the IPv6 network.

4.1. Rationale

This document introduces two new functional elements (Figure 1):

1. The mAFTR: responsible for replicating IPv4 multicast flows in the IPv6 domain owing to a stateless IPv4-in-IPv6 encapsulation function. The mAFTR does not undertake any NAT operation. The mAFTR is a demarcation point which connects to both the IPv4 and IPv6 multicast networks.

2. The mB4: is a functional entity embedded in a CPE responsible for the de-capsulation of the received IPv4-in-IPv6 multicast packets and forwarding them to the appropriate IPv4 receivers.

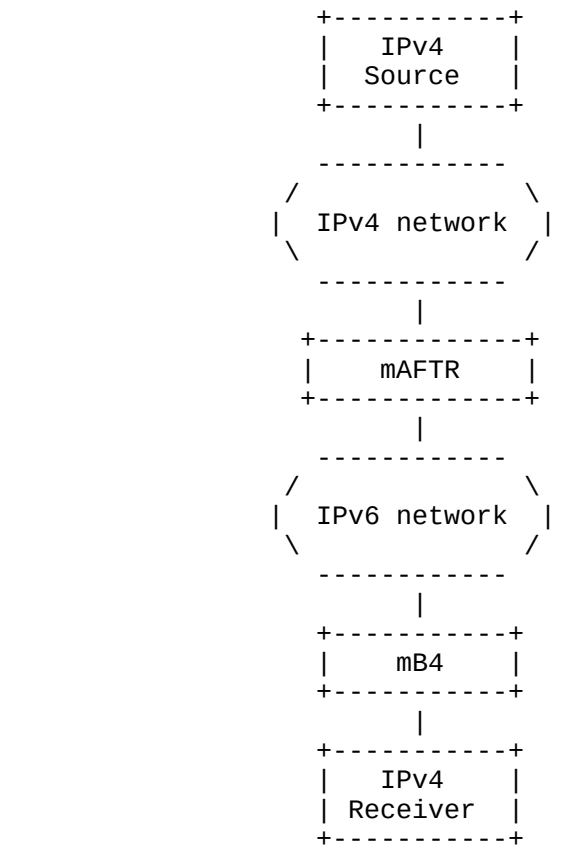


Figure 1: Functional Architecture

4.2. IPv4-embedded IPv6 Address Prefixes

A dedicated IPv6 multicast prefix (mPrefix64) is needed for forming IPv6 multicast addresses, with IPv4 multicast address embedded. The mPrefix64 can be of two types: ASM_mPrefix64 (an mPrefix64 used in ASM mode) or SSM_mPrefix64 (an mPrefix64 used in SSM mode), and MUST be derived from the corresponding IPv6 multicast address space [I-D.boucadair-behave-64-multicast-address-format].

In addition, the address of the IPv4 multicast source should be mapped to IPv6 addresses in the IPv6 realm: an IPv6 unicast prefix (uPrefix64) is therefore needed for forming IPv6 unicast addresses with IPv4 unicast address embedded. The uPrefix64 MUST be derived from the IPv6 unicast address space [RFC6052].

The mAFTR and mB4 MUST use the same mPrefixe64 and uPrefix64, and the

same algorithm for building IPv4-embedded IPv6 addresses. Refer to Section 5 for more details on the IPv6 address format.

4.3. Multicast Distribution Tree

Assume that an IPv4 receiver sends an IGMP Report towards the mB4 to join a given multicast group. After receiving the IGMP Report message, the mB4 converts the IGMP message into a MLD Report [RFC2710] message which will then be forwarded upstream towards the MLD Querier. The MLD Querier is likely to coexist with the PIM DR where the PIMv6 Join message will be triggered and sent up hop by hop along the PIMv6 routers. Note that the mAFTR is in the path to reach the IPv4 source; this is typically achieved by the underlying unicast IPv6 routing protocol that advertises the unicast IPv4-embedded IPv6 addresses: these addresses are used to represent IPv4 sources in the IPv6 multicast domain.

Both the MLD and the PIMv6 Join messages convey the IPv6 address of the multicast group to be joined. The corresponding IPv6 multicast group address is constructed by using the pre-configured mPrefix64 and an algorithm so that the IPv4 multicast group address is embedded accordingly.

When source-specific multicast is deployed, the IPv6 address of the multicast source should be constructed in the same way (using uPrefix64, with IPv4 multicast source embedded). Refer to Section 6.1 for more details of the mB4 function.

- If the mAFTR is embedded in the MLD Querier/PIMv6 DR, it should process the received MLD Report message for the IPv4-embedded IPv6 group and send the corresponding IPv4 PIM Join message.

- If the mAFTR is embedded in some upstream PIMv6 router more than one hop away from the mB4, it should process the received PIMv6 Join message for the IPv4-embedded IPv6 group and send the corresponding IPv4 PIM Join message.

In both cases, an entry for an IPv6 multicast group address is created by the mAFTR in its multicast Routing Information Base and is used to forward multicast IPv4-in-IPv6 datagrams. Refer to Section 7.1 for more details about the mAFTR function.

A branch of the multicast distribution tree is then established, comprising both an IPv4 part (from the mAFTR upstream) and an IPv6 part (between the mB4 and the mAFTR).

4.4. Multicast Forwarding

Whenever an IPv4 multicast packet is received on a mAFTR (this assumes the RPF Check has passed Section 7.1), it will be encapsulated into an IPv6 packet using the IPv4-embedded IPv6 multicast address as the destination address and an IPv4-embedded IPv6 unicast address as the source of the IPv4-in-IPv6 packet. The new IPv6 multicast packet will then be sent through the outgoing interface of the matching entry in the multicast routing table and forwarded down the IPv6 multicast distribution tree towards the mB4.

When receiving the packet, the mB4 should de-capsulate it and forward the original IPv4 multicast packet to the appropriate receiver. If mB4 does not have any route to forward the packet (e.g., change of the IPv4 address without cleaning MLD states), the encapsulated IPv4 datagram is silently dropped.

Note that: There is an alternative to the encapsulation based mechanism (which is detailed in this memo) for Multicast Forwarding: Translation based approach, which is per [I-D.boucadair-behave-64-multicast-address-format], [RFC6052] and [RFC6145]. Refer to Appendix A.

4.5. Multicast DS-Lite vs. Unicast DS-Lite

Unlike a unicast AFTR, a mAFTR does not perform any NAT for delivering IPv4 multicast traffic.

Unlike unicast DS-Lite, a mB4 does not need to discover a mAFTR.

mAFTR is responsible for encapsulating in a stateless manner the IPv4 multicast traffic into IPv6 datagrams. mB4 is responsible for de-capsulating in a stateless manner the IPv4-in-IPv6 multicast traffic. Further elaboration is provided in the following sections about the behaviour of the mAFTR and the mB4.

The corresponding multicast DS-Lite and the unicast DS-Lite functional elements can be co-located in the same device or separated.

5. Address Mapping

5.1. Prefix Assignment

In order to map the addresses of IPv4 multicast traffic with IPv6 multicast addresses, an IPv6 multicast prefix (mPrefix64) and an IPv6 unicast prefix (uPrefix64) are provided to mAFTR and mB4 elements.

The address format to be used is being left to the responsibility of the service provider as indicated in [RFC6052] and [I-D.boucadair-behave-64-multicast-address-format].

The mPrefix64 and uPrefix64 together with the address format to be used can be configured in the mB4 through a dedicated provisioning protocol, such as DHCPv6 or another protocol. Two candidate DHCPv6 options are identified in [I-D.ietf-behave-nat64-learn-analysis].

5.2. Text Representation Examples

Group address mapping example when a /96 is used:

mPrefix64	IPv4 address	IPv4-Embedded IPv6 address
ffxx:abc::/96	230.1.2.3	ffxx:abc::230.1.2.3

Source address mapping example when a /96 is used:

uPrefix64	IPv4 address	IPv4-Embedded IPv6 address
2001:db8::/96	192.1.2.3	2001:db8::192.1.2.3

6. Multicast B4 (mB4)

6.1. IGMP-MLD Interworking function

IGMP-MLD Interworking function combines the IGMP/MLD Proxying function specified in [RFC4605] and the IGMP/MLD adaptation function which is meant to reflect the contents of IGMP messages into MLD messages.

Then mB4 performs the router portion of the IGMP protocol on each downstream interface and performs the host portion of the MLD protocol on the upstream interface (Figure 2).

The output of the operation is a set of membership information which is maintained separately on each downstream interface (e.g., Wifi and Wired Ethernet). In addition, the membership information on each downstream interface is merged into the membership database on which the IPv4 multicast packets are forwarded by mB4.

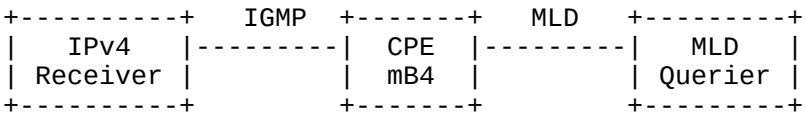


Figure 2: IGMP-MLD Interworking

When an IGMP Report message is received from a receiver to subscribe to a given multicast group G (e.g., 230.1.2.3) (and optionally associated to a source 192.1.2.3 if SSM mode is used), the mB4 MUST send an MLD Report message to subscribe to the corresponding IPv6 group identified by an IPv4-embedded IPv6 multicast address using a pre-configured prefix and algorithm (e.g., ffxx:abc::230.1.2.3 (and optionally source 2001:db8::192.1.2.3 if SSM mode is used)). The MLD Report message is sent through the upstream interface natively (i.e., without any encapsulation).

6.2. De-capsulation and Forwarding

When the mB4 receives an IPv6 multicast packet, it checks whether the group address is in the range of mPrefix64 and the source address is in the range of uPrefix64. If it is true, the mB4 MUST de-capsulate the IPv4-in-IPv6 packets to extract the original IPv4 multicast packets.

Then the IPv4 multicast packet will be forwarded to downstream receivers based on information maintained by the mB4 in the membership database. If no route is found, the packet is silently dropped.

6.3. Fragmentation

Encapsulating IPv4 over IPv6 from mAFTR to mB4 for data forwarding reduces the effective MTU size by the size of an IPv6 header (assuming [RFC2473] encapsulation). To avoid fragmentation, a service provider may increase the MTU size by 40 bytes on the IPv6 network or mAFTR and mB4 may use IPv6 Path MTU discovery.

6.4. Host with mB4 function embedded

The mB4 function can be embedded in the CE or in a dual-stack host behind the CP router (e.g., STB). If mB4 is embedded in the STB, the IGMP-MLD interworking function is not needed. The STB should formulate the MLD message correspondingly based on given IPv4 group address to be joint using mPrefix64 (and uPrefix64 for IPv4-embedded source if SSM is deployed), and de-encapsulate the downstream multicast traffics received by itself.

7. Multicast AFTR (mAFTR)

7.1. Routing Considerations

Except the need for the mAFTR to belong to IPv4 multicast distribution trees and to be on the reverse path towards the source when performing RPF checks on PIMv6 routers, no further routing constraint is to be taken into account.

Having the mAFTR in the reverse path ensures PIM Join sent to the source (e.g., SSM mode or SPT mode in ASM) will be intercepted by the mAFTR.

7.2. Processing PIM/MLD Join Messages

Upon receipt of the PIM/MLD Join for an IPv6 group (e.g., ffxx:abc::230.1.2.3), the mAFTR checks the corresponding entry in the IPv6 multicast routing table and adds the IPv6 interface through which the Join message has been received into the Out-Interface-List of that entry. If the entry does not exist, a new one will be created, as per typical PIM machinery [RFC4601]. The mAFTR should check whether the IPv6 group address belongs to the mPrefix64 (e.g., ffxx:abc::/96). If so, the mAFTR will need to extract the IPv4 group address (e.g., 230.1.2.3) from the IPv4-embedded IPv6 address (e.g., according to [I-D.boucadair-behave-64-multicast-address-format]) and check the corresponding entry in the IPv4 multicast routing table then add the tunnel interface into the Out-Interface-List of that entry. If the entry does not exist, a new entry should be created and a PIM join message for that IPv4 group will be sent towards the RP or source connected to the IPv4 network.

When SSM is deployed, the mAFTR would in addition check if the source (e.g., 2001:db8::192.1.2.3) described in the PIMv6 Join message belongs to uPrefix64 (e.g., 2001:db8::/96). If so, it can then send a PIM (S, G) Join message directly towards the IPv4 source (e.g., 192.1.2.3).

The initialization of the tunnel interface (used for encapsulation purposes) on the mAFTR is out of the scope of this document.

7.3. Reliability

For robustness, reliability and load distribution purposes, several nodes in the network can embed the mAFTR function. In such case, the same IPv6 prefixes (i.e., mPrefix64 and uPrefix64) and algorithm to build IPv4-embedded IPv6 addresses MUST be configured on those nodes.

7.4. ASM Mode: Building Shared Trees

7.4.1. IPv4 Side

For a given Rendezvous Point (RP) used in the IPv4 realm, there is no new requirement. Like any other IPv4 PIM router, the RP of each IPv4 multicast groups is configured to the mAFTR or discovered using some appropriate means. Moreover, PIM-SM registration procedure [RFC4601] in the IPv4 realm is not impacted.

Shared IPv4 multicast trees are built using the procedure defined in [RFC4601] for instance.

7.4.2. IPv6 Side

In the IPv6 side, the RP of IPv4-embedded IPv6 multicast groups is configured to all IPv6 PIM routers or discovered using appropriate means. For the sake of simplicity, it is RECOMMENDED to configure an mAFTR as the RP for IPv4-embedded IPv6 multicast groups.

[Note 1: If some other IPv6 multicast router wants to become the RP of the IPv4-embedded IPv6 multicast groups, it may require an mAFTR to emulate the PIM Source Register procedure on behalf of IPv4-embedded IPv6 sources with the RP. The PIM Source Register procedure in the IPv4 domain is not altered.]

[Note 2: How the mAFTR is aware about the sources? This can be considered as deployment-specific:

(i) By configuration: mAFTR can be configured to join a set of IPv4 multicast groups and to initiate a registration procedure on behalf of a set of sources to the RP in the v6 domain;

(ii) Dynamic: this assumes that mAFTR is configured to join a set of IPv4 multicast groups. The source address of received flows will be used as a trigger to initiate the registration procedure to the RP in the IPv6 domain. There is a special case where mAFTR is the RP of the IPv4 group in the IPv4 domain: The registration procedure should then be relayed to the RP in the IPv6 domain.

]

Shared IPv6 multicast trees are built using the procedure defined in [RFC4601] for instance. Switching from a shared tree to source-based tree can be accommodated since the mAFTR is in the path to join the source.

The mAFTR will graft to the IPv4 shared tree either because it has been configured with the list of IPv4 multicast groups that will be subscribed by the DS-Lite serviced receivers downstream or upon receipt of a PIMv6 Join message.

An example of the exchange of PIM messages is illustrated in Figure 3.

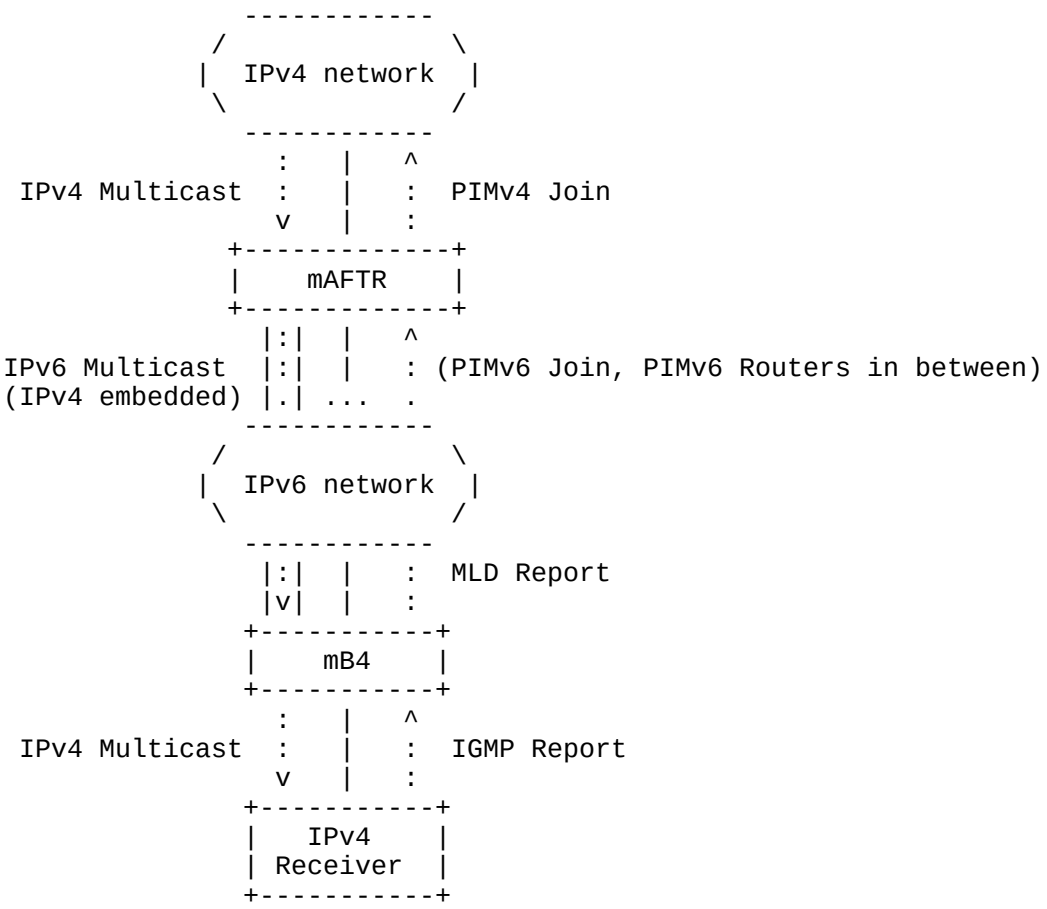


Figure 3: Procedure Overview

7.5. TTL/Scope

The Scope field of IPv4-in-IPv6 multicast addresses can be valued to "E" (Global scope) or to "8" (Organization-local scope). This is left to service providers taste.

7.6. Encapsulation and forwarding

When receiving an IPv4 multicast packet, a lookup of the IPv4 multicast routing table is performed by the PIMv4 router that embeds the mAFTR capability. If an interface used for IPv4-in-IPv6 encapsulation is found in the Out-Interface-List of the matching entry, the encapsulation operation is triggered. The mAFTR encapsulates in a stateless fashion the IPv4 multicast packet into an IPv6 multicast datagram. It MUST use the pre-provisioned mPrefix64/ uPrefix64 together with an algorithm for building the IPv4-embedded IPv6 multicast address that identifies the multicast group, as well as the IPv6 source address that represents the IPv4 source in the IPv6 network.

As an illustration, if a packet is received from source 192.1.2.3 and forwarded to group 230.1.2.3, the mAFTR encapsulates it into an IPv6 multicast packet using ffxx:abc::230.1.2.3 as the destination IPv6 address and 2001:db8::192.1.2.3 as the multicast source address.

Then a lookup of the IPv6 multicast routing table is performed by the PIMv6 router that embeds the mAFTR capability, based on the IPv4-embedded IPv6 address. If a matching entry is found and there exist IPv6 interfaces in the Out-Interface-List, the IPv6 multicast packet will be sent out through these interfaces and forwarded down the multicast distribution tree towards the mB4 devices.

8. Optimization in L2 Access Networks

The approach specified in this document is compatible with a Layer-2 infrastructure which may be involved for deterministic multicast replication.

The IPv4-in-IPv6 encapsulated multicast flows destined to IPv4-embedded IPv6 group addresses are treated as any IPv6 multicast flow, and can be replicated across Multicast VLANs. Additionally, mechanisms such as MLD Snooping, MLD Proxying, etc., can be introduced into the distributed Access Network Nodes (e.g., Aggregation Switches, xPON devices) which then could behave as MLD Querier and replicate multicast flows as appropriate. Thus, the multicast replication point is moved downward closer to the receivers, so that bandwidth consumption is optimized.

9. Security Considerations

This document does not introduce any new security concern in addition to what is discussed in Section 5 of [RFC6052], Section 10 of

[RFC3810] and Section 6 of [RFC4601].

9.1. Firewall Configuration

The CPE should be configured to accept incoming MLD messages and traffic forwarded to multicast groups subscribed by receivers located in the customer premises.

10. Acknowledgements

The authors would like to thank Dan Wing for his guidance in the early discussions which initiated this work. We also appreciate Peng Sun, Jie Hu, Qiong Sun, Lizhong Jin, Alain Durand, Dean Cheng, and Behcet Sarikaya for their valuable comments.

11. IANA Considerations

This document includes no request to IANA.

12. References

12.1. Normative References

[I-D.boucadair-behave-64-multicast-address-format]
Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X., and M. Xu, "IPv4-Embedded IPv6 Multicast Address Format", draft-boucadair-behave-64-multicast-address-format-01 (work in progress), February 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.

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

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.

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

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

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

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

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

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

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

12.2. Informative References

[I-D.ietf-behave-nat64-learn-analysis]
Korhonen, J. and T. Savolainen, "Analysis of solution proposals for hosts to learn NAT64 prefix", draft-ietf-behave-nat64-learn-analysis-00 (work in progress), May 2011.

[I-D.ietf-mboned-multiaaa-framework]
Satou, H., Ohta, H., Hayashi, T., Jacquenet, C., and H. He, "AAA and Admission Control Framework for Multicasting", draft-ietf-mboned-multiaaa-framework-12 (work in progress), August 2010.

[I-D.jaclee-behave-v4v6-mcast-ps]
Jacquenet, C., Boucadair, M., Lee, Y., Qin, J., and T. ZOU), "IPv4-IPv6 Multicast: Problem Statement and Use Cases", draft-jaclee-behave-v4v6-mcast-ps-02 (work in progress), June 2011.

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.

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

[RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, August 2006.

[RFC4608] Meyer, D., Rockell, R., and G. Shepherd, "Source-Specific Protocol Independent Multicast in 232/8", BCP 120, RFC 4608, August 2006.

Appendix A. Translation vs. Encapsulation

In order to deliver IPv4 multicast flows to DS-Lite serviced receivers, two options can be considered:(1) Translation; (2)Encapsulation.

It should be noted that some contract agreement may prevent the contents from being altered. In this case, the employment of the translation approach may raise issues e.g., Integrity Check failures.

A.1. Translation

To delivery IPv4 multicasst contents to an IPv4 receiver: Introduce translation functions at the boundaries of IPv6 network. The IPv4-translated multicast streams are distributed within the IPv6 network natively until the customer premises device where the IPv4-translated IPv6 streams are translated back and passed to IPv4 receivers. Multicast Distribution Tree is established by normal machinery of control protocols (e.g. IGMP, MLD, PIMv4/v6) and the Interworking functions (e.g. IGMP-MLD, PIMv6-PIMv4), refer to Section 6 and Section 7. The translation function is stateless owing to the use of IPv4-Embedded IPv6 address [I-D.boucadair-behave-64-multicast-address-format] and [RFC6052].

A.2. Encapsulation

To deliver IPv4 multicast contents to an IPv4 receiver: Introduce two elements at the boundaries of IPv6 network, mAFTR and mB4. Multicast Distribution Tree is established by normal machinery of control protocols (e.g. IGMP, MLD, PIMv4/v6) and the Interworking functions (e.g. IGMP-MLD, PIMv6-PIMv4), refer to Section 6 and Section 7. Multicast streams are forwarded to a receiver by using an IPv4-in-IPv6 encapsulation scheme. The encapsulation/de-capsulation function is stateless owing to the use of IPv4-Embedded IPv6 address [I-D.boucadair-behave-64-multicast-address-format] and [RFC6052].

Authors' Addresses

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

Phone: +86 10 5855 2177
Email: wangqian@ctbri.com.cn

Jacni Qin
ZTE
Shanghai,
China

Phone: +86 1391 8619 913
Email: jacniq@gmail.com

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Phone:
Email: mohamed.boucadair@orange-ftgroup.com

Christian Jacquenet
France Telecom
Rennes, 35000
France

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

Yiu L. Lee
Comcast
U.S.A.

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

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

B. Sarikaya
Huawei USA

March 7, 2011

Multicast Support for Dual Stack Lite and 6RD
draft-sarikaya-softwire-dslite6rdmulticast-00.txt

Abstract

This memo specifies modifications required to DS-Lite and 6RD so that both IPv4/ IPv6 hosts can receive multicast data from IPv4/ IPv6 servers.

The DS-Lite solution is based on DS-Lite Basic Bridging BroadBand element (B4) proxying IGMP and then tunneling IGMP messages to DS-Lite Address Family Transition Router element (AFTR). IPv4 multicast data received at AFTR is tunneled to B$ and then delivered to the hosts. IPv6 multicast and MLD can be supported in a similar way.

The 6RD protocol is based on proxying MLD at the 6RD Customer Edge and then tunneling MLD messages to 6RD Border Relays where IPv6 multicast routing is supported. Multicast data received at 6RD Border Relay is tunneled to 6RD Customer Edge node and then delivered to the hosts. We show that IPv4 multicast and IGMP can be supported in a similar way.

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 8, 2011.

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

With IPv4 address depletion on the horizon, many techniques are being standardized for IPv6 migration including DS-Lite [I-D.ietf-softwire-dual-stack-lite] and 6RD [RFC5969]. DS-Lite enables IPv4 hosts to communicate with external hosts using IPv6 only network and moves the traditional NAT to the network. B4 element's LAN side is dual stack and WAN side is IPv6 only. B4 tunnels IPv4 packets received from the LAN side to AFTR element after encapsulating IPv4 packet in an IPv6 packet. AFTR decapsulates the packet, does a NAT operation and then sends the packet out to IPv4 public internet.

6RD enables IPv6 hosts to communicate with external hosts using IPv4 only legacy ISP network. 6RD Customer Edge (CE) device's LAN side is dual stack and WAN side is IPv4 only. CE tunnels IPv6 packets received from the LAN side to 6RD Border Relays (BR) after encapsulating IPv6 packet in an IPv4 packet. BRs have anycast IPv4 addresses and receive encapsulated packets from CEs over a virtual interface. 6RD operation is stateless. Packets are received/ sent independent of each other and no state needs to be maintained.

DS-Lite as defined in [I-D.ietf-softwire-dual-stack-lite] is unicast only, it does not support multicast. In this document we specify how multicast from home IPv4 users can be supported in DS-Lite. We also show how IPv6 multicast can be supported for home IPv6 users in DS-Lite.

6RD as defined in [RFC5969] and [RFC5569] is unicast only. It does not support multicast. In this document we specify how multicast from home IPv6 users can be supported in 6RD. We also show how IPv4 multicast can be supported for home IPv4 users. Both solutions use IPv6 and IPv4 multicast addressing and do not require any new multicast address prefixes such as IPv4-embedded IPv6 multicast addresses to be allocated.

2. Terminology

This document uses the terminology defined in [RFC5969], [RFC5569], [RFC3810] and [RFC3376].

3. Requirements

This section states requirements on DS-Lite and 6RD multicast support protocol.

DS-Lite B4 MUST support IGMP Proxy as defined in [RFC4605]. DS-Lite B4 MAY support MLD Proxy.

DS-Lite AFTR MUST support IGMP Querrier. DS-Lite AFTR MAY support MLD Querrier.

6RD CE MUST support MLD Proxy as defined in [RFC4605]. 6RD CE MAY support IGMP Proxy.

6RD BR MUST support MLD Querrier. 6RD CE MAY support IGMP Querrier.

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

4. Architecture

In DS-Lite, there are hosts (possibly IPv4/ IPv6 dual stack) served by B4 element. B4 is dual stack facing the hosts and IPv6 only facing the network or WAN side. At the boundary of the network there is AFTR. AFTR receives IPv4 packets tunneled in IPv6 from B4 and decapsulates them and sends them out to IPv4 network.

In order to support multicast B4 implements IGMP Proxy function [RFC4605]. IPv4 hosts send their join requests (IGMP Membership Report messages) to B4. B4 as a proxy sends aggregated Report messages upstream towards AFTR.

AFTR is the default multicast querier for B4. AFTR implements multicast router function or it could be another IGMP proxy.

All the elements of DS-Lite multicast support system are shown in Figure 1.

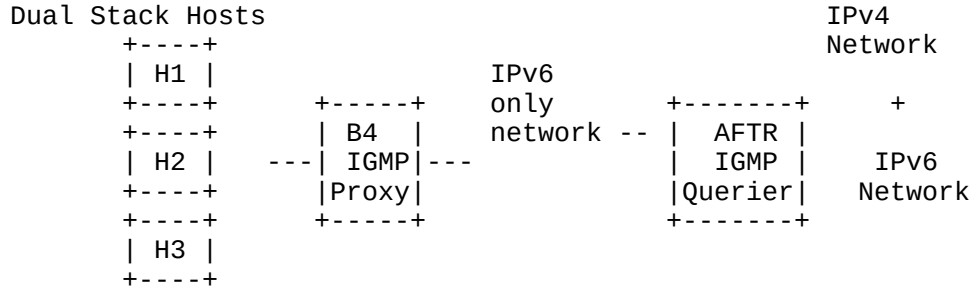


Figure 1: Architecture of DS-Lite Multicast Protocol

In 6RD, there are hosts (possibly IPv4/ IPv6 dual stack) served by 6RD Customer Edge device. CE is dual stack facing the hosts and IPv4 only facing the network or WAN side. At the boundary of the network there is 6RD Border Relay. BR receives IPv6 packets tunneled in IPv4 from CE and decapsulates them and sends them out to IPv6 network.

In order to support multicast CE implements MLD Proxy function [RFC4605]. IPv6 hosts send their join requests (MLD Membership Report messages) to CE. CE as a proxy sends aggregated Report messages upstream towards BR.

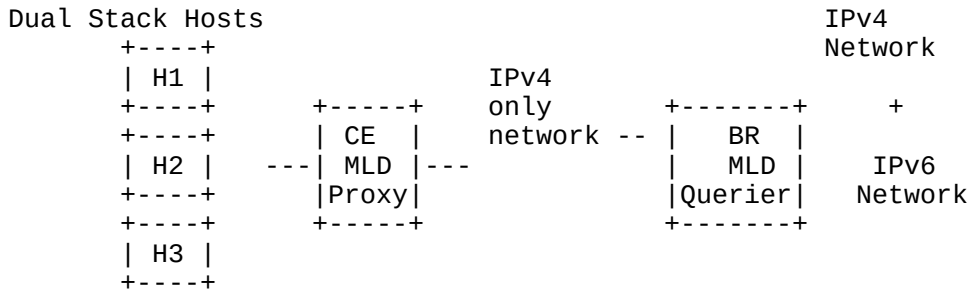


Figure 2: Architecture of 6RD Multicast Protocol

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

All the elements of 6RD multicast support system are shown in Figure 2.

5. DS-Lite Multicast Operation

In this section we specify how the host can subscribe and receive IPv4 multicast data from IPv4 content providers based on the architecture defined in Section 4.

The hosts will send their subscription requests for IPv4 multicast groups upstream to the default router, i.e. B4 Element. After subscribing to the group, the host can receive multicast data from the B4. The host implements IGMP protocol’s host part.

B4 Element is IGMP Proxy. After receiving the first IGMP Report message requesting subscription to an IPv4 multicast group, B4 establishes a tunnel interface with a AFTR. The tunnel is IPv6 based but it will carry IP traffic, IGMP messages back and forth and IPv4 multicast data messages downstream. This is similar to [I-D.ietf-multimob-pmipv6-base-solution] but the operation is much simpler. In DS-Lite environment there is no requirement to handle host mobility. B4 does not have to keep more than one tunnel interfaces, a single interface is sufficient. IGMP Proxy at the B4 does not have to have more than one proxy instances, a single instance is sufficient.

B4 is regular IGMP proxy and it keeps IGMP proxy membership database. B4 inserts multicast forwarding state on the incoming interface, and merges state updates into the IGMP proxy membership database. B4 updates or removes elements from the database as required. B4 will then send an aggregated Report via the upstream tunnel to the AFTR when the membership database changes.

B4 answers IGMP queries from AFTR based on the membership database. B4’s downstream link follows the traditional multipoint channel forwarding and does not pose any specific problems.

B4 receives IPv4 multicast data from the AFTR tunneled over the tunnel interface. B4 decapsulates the packet and then forwards it downstream. Each member host receives the data packet based on Layer 2 multicast interface. No packet duplication is necessary.

AFTR acts as the as the default multicast querier for all B4s that have established an IPv6 tunnel with it. In order to keep a consistent multicast state between a B4 and AFTR, once a B4 is connected it will stay connected until the state becomes empty. After that point, the B4 may continue to use the tunnel for IPv4 unicast traffic.

According to aggregated IGMP reports received from a B4, AFTR establishes group/source-specific multicast forwarding states at its corresponding downstream tunnel interfaces. After that, AFTR maintains or removes the state as required by the aggregated reports received from B4.

At the upstream interface, AFTR procures for aggregated multicast membership maintenance. Based on the multicast-transparent operations of the B4s, the AFTR treats its tunnel interfaces as multicast enabled downstream links, serving zero to many listening nodes.

Multicast traffic arriving at the AFTR is transparently forwarded

according to its multicast forwarding information base. Multicast data is first replicated and then forwarded in IPv4-in-IPv6 tunnel from AFTR to the corresponding B4.

5.1. Tunnel Interface Considerations

Legacy IPv4 in IPv6 tunneling is performed as in [RFC2473]. Considerations specified in [I-D.ietf-softwire-dual-stack-lite] apply. Packets upstream from B4 carry only IGMP signaling messages and they are not expected to fragmentation. However packets downstream, i.e. multicast data to B4 may be subject to fragmentation.

5.2. Supporting IPv6 Multicast in DS-Lite

IPv6 multicast can be supported in a way similar to IPv4 as described in Section 5. B4 Element has MLD Proxy function. Proxy operation for MLD [RFC3810] is described in [RFC4605].

B4 receives MLD join requests from the hosts and then sends aggregated MLD Report messages upstream in an IPv6 in IPv6 tunnel. Tunnel addressing is in IPv6 and is as described in [I-D.ietf-softwire-dual-stack-lite]. Multicast membership database is maintained for all active IPv6 multicast groups the hosts subscribe.

AFTR is MLD querier or another MLD Proxy. It serves all B4s downstream and treats its tunnel interfaces as multicast enabled downstream links, serving zero to many listening nodes. Multicast membership database is maintained based on the aggregated Reports received from downstream tunnel interfaces.

IPv6 multicast data received from the multicast Single Source Multicast or Any Source Multicast sources are replicated according to the multicast membership database and the data packets are tunneled to the B4s that have one or more members of this multicast group.

B4s receive multicast data upstream in the B4-AFTR tunnel and decapsulate it and then forward the packet downstream. Each member host receive IPv6 multicast data packet from its Layer 2 interface.

6. 6RD Multicast Operation

In this section we specify how the host can subscribe and receive IPv6 multicast data from IPv6 content providers based on the architecture defined in Section 4.

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

Customer Edge device is MLD Proxy. After receiving the first MLD Report message requesting subscription to an IPv6 multicast group, CE establishes a tunnel interface with a Border Relay. The tunnel is IPv4 based but it will carry IP traffic, MLD messages back and forth and IPv6 multicast data messages downstream. This is similar to [I-D.ietf-multimob-pmipv6-base-solution] but the operation is much simpler. In 6RD environment there is no requirement to handle host mobility. CE does not have to keep more than one tunnel interfaces, a single interface is sufficient. MLD Proxy at the CE does not have to have more than one proxy instances, a single instance is sufficient.

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

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

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

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

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

At the upstream interface, BR procures for aggregated multicast membership maintenance. Based on the multicast-transparent

operations of the CEs, the BR treats its tunnel interfaces as multicast enabled downstream links, serving zero to many listening nodes.

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

6.1. Tunnel Interface Considerations

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

6.2. Supporting IPv4 Multicast in 6RD

IPv4 multicast can be supported in a way similar to IPv6 as described in Section 6. 6RD Customer Edge device has IGMP Proxy function. Proxy operation for IGMP [RFC3376] is described in [RFC4605].

CE receives IGMP join requests from the hosts and then sends aggregated IGMP Report messages upstream in an IPv4 in IPv4 tunnel. Tunnel addressing is in IPv4 and is as described in [RFC5969]. Multicast membership database is maintained for all active IPv4 multicast groups the hosts subscribe.

6RD Border Relay is IGMP querier or another IGMP Proxy. It serves all CEs downstream and treats its tunnel interfaces as multicast enabled downstream links, serving zero to many listening nodes. Multicast membership database is maintained based on the aggregated Reports received from downstream tunnel interfaces.

IPv4 multicast data received from the multicast Single Source Multicast or Any Source Multicast sources are replicated according to the multicast membership database and the data packets are tunneled to the CEs that have one or more members of this multicast group.

CEs receive multicast data upstream in the CE-BR tunnel and decapsulate it and then forward the packet downstream. Each member host receive IPv4 multicast data packet from its Layer 2 interface.

7. Security Considerations

TBD.

8. IANA Considerations

TBD.

9. Acknowledgements

TBD.

10. References

10.1. Normative References

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

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

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

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

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

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

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

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

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

[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-07 (work in progress), March 2011.

[I-D.ietf-multimob-pmipv6-base-solution]
Schmidt, T., Waehlisch, M., and S. Krishnan, "Base Deployment for Multicast Listener Support in PMIPv6 Domains", draft-ietf-multimob-pmipv6-base-solution-07 (work in progress), December 2010.

10.2. Informative references

Author’s Address

Behcet Sarikaya
Huawei USA
1700 Alma Dr. Suite 500
Plano, TX 75075

Phone: +1 972-509-5599
Email: sarikaya@ieee.org

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: August 2, 2011

T. Tsou
Huawei Technologies (USA)
C. Zhou
Huawei Technologies
H. Ji
China Telecom
January 29, 2011

A Generic Approach to Multicast Encapsulation In Support of IPv6 Transition
draft-tsou-softwire-encapsulated-multicast-00

Abstract

Consider a situation which will arise in many IPv6 transition scenarios, where Network A, to which a host is attached, supports one IP version, but the host and Network B support a different IP version. Suppose that the host wishes to access a multicast group which is rooted or sourced in Network B. This document specifies an approach that combines stateful translation for signalling, encapsulation of multicast content moving between Network B and the host, and native multicast routing in Network A to provide the host with its desired access.

Status of this Memo

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

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

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

This Internet-Draft will expire on August 2, 2011.

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

Transition scenarios have been explored in which an IPv6 host attached to an IPv4 network wishes to access content in an IPv6 network, or conversely, an IPv4 host attached to an IPv6 network wishes to access content in an IPv4 network. A long list of tools has been put forward for passing unicast content across the network in the middle based on tunneling.

Some work has also been done on conveying multicast streams between IPv4 and IPv6 networks, in either direction. Of particular interest is current work in [ID.softwire-dslite-multicast]. However, the present document differs from [ID.softwire-dslite-multicast] both in its degree of generality and in the detailed mechanism used for translation between IPv4 and IPv6 multicast addresses. The present document does not restrict operation to specially constructed IPv6 multicast addresses. Instead it makes use of the fact that for a given network, it is unnecessary to map the complete universe of IPv6 addresses into IPv4, but only those addresses actually being carried through the network.

This document is adapted from [ID.behave-translated-multicast] and uses the same basic mechanism. It requires additional bandwidth because of its use of encapsulation for the multicast content, but thereby avoids the need to translate the addressing of that content between IPv4 and IPv6 at the receiving end of the tunnel.

1.1. Requirements Language

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

2. Problem Description

We consider, as described in the previous section, a host supporting one IP version, say IPvx, attached to a provider network supporting a different version, say IPvy. Obviously there has to be an adaptation function between the host and the network to make this work. This document assumes that the adaptation function for unicast packets consists of tunneling combined with a suitable choice of destination address to steer the packets to the right border gateways.

On the other side of the provider network, border gateways connect to neighbouring networks. If a particular neighbouring network supports a different version of IP -- that is, IPvx, then the border gateway must also implement adaptation functions. In particular, the unicast

adaptation function at the border gateway is complementary to the adaptation function at the host side.

Multicast streams could simply be tunneled from the border to the host. However, to save bandwidth, it is desirable to use the native multicast capabilities of the IPvy network so that paths can be shared as much as possible. This implies three requirements on the multicast adaptation function:

- o it has to enable the use of multicast signalling to build distribution trees in the IPvy network;
- o it has to route multicast content through those distribution trees rather than directly across the network;
- o by specific assumption of this document, it must encapsulate incoming IPvx packets before forwarding them to the distribution trees, and decapsulate outgoing packets before sending them onwards.

The basic situation just described is illustrated in Figure 1. The host-side adaptation functions MAY be implemented in the host itself, in a separate piece of equipment at the customer site (CPE-based approach), or at the provider edge (gateway initiated approach). The border adaptation functions MUST be implemented in border gateways.

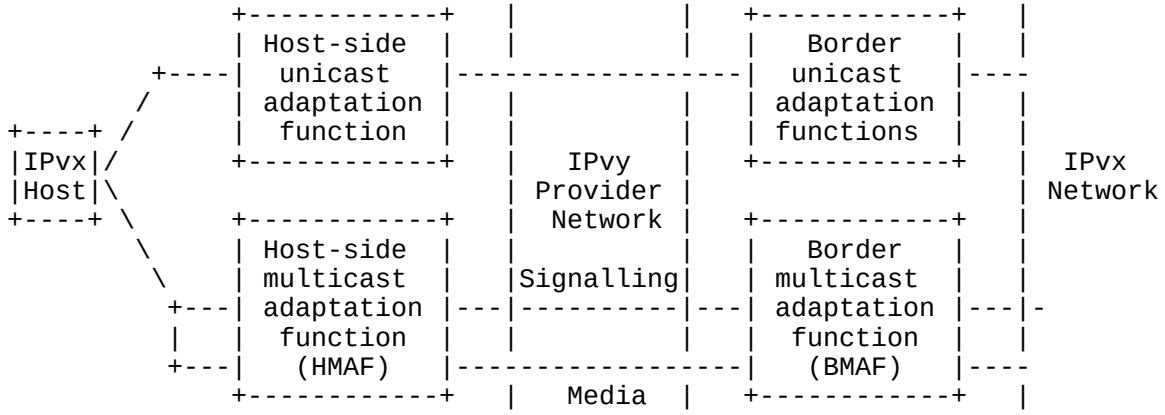


Figure 1: Adaptation Functions For Flows Crossing Two IP Version Boundaries

The key assumption of this document is that when the host wishes to acquire a multicast stream rooted or sourced in the IPvx network, it knows only the IPvx address pair <Source, Group> (where the source

MAY be wild-carded, i.e., for an any-source multicast group).

It learns that address pair by means outside the scope of this specification (e.g., via the web or session signalling).

As a result, for purposes of multicast signalling, the host-side multicast adaptation function (HMAF) needs to obtain a mapping between this IPvx address pair and the corresponding IPvy address pair used in the IPvy network to denote the same multicast stream. Similarly, the border multicast adaptation function (BMAF) needs this mapping both for purposes of multicast signalling and so that it can assign the right IPvy source and destination addresses to incoming IPvx multicast content.

3. Proposed Solution

The proposed solution consists of three elements:

- a stateful mapping function within the IPvy provider network that provides mappings between IPvx <Source, Group> address pairs and corresponding IPvy <Source, Group> address pairs denoting the same multicast flows;
- address pools of IPvy multicast and unicast addresses provisioned at the mapping function;
- a protocol that allows the HMAF and BMAF to request mappings from the mapping function. PCP [ID.port-control-protocol] is a candidate for this protocol, but that decision needs further consideration.

3.1. How It Works

1. Initial discovery and Join request

The IPvx host discovers the <Source, Group> address pair of a multicast stream the user wants to receive. The IPvx Host sends an MLDv2 [RFC3810] (for IPv6) or IGMPv3 [RFC3376] (for IPv4) Join request to the HMAF to acquire the stream.

2. <Source, Group> Address Mapping At the HMAF

The HMAF checks its cache of mappings to see if it already has a mapping between the IPvx <Source, Group> address pair received in the host request and a corresponding pair of IPvy addresses. Failing to find a mapping, it sends a request for the required mapping to the mapping function. The mapping function in turn checks whether it has

already created the mapping. If not, it assigns unicast and multicast IPvy addresses from its pool and records the mapping for further use. In either case it returns the requested mapping to the HMAF, which caches it. [Editor's Note: The transaction is carried out over a protocol to be specified in a later version of this document.]

3. Propagation Of the Join Request Into the IPvy Network

Using the mapping it has received, the HMAF interworks from MLDv2 to IGMPv3 or vice versa, depending on whether the host supports IPv6 or IPv4. It forwards the interworked Join request to the Provider IP Edge.

If the HMAF is collocated with the Provider IP Edge, this interworking step is an internal operation.

The Provider IP Edge acts on the received request by interworking it to a Protocol Independent Multicast - Sparse Mode (PIM-SM) [RFC4601] request and forwarding that request into the IPvy network, indicating the IPvy <Source, Group> address pair it was given and ensuring that it is on the multicast tree for the stream concerned.

Assuming that the multicast tree for the requested stream is not joined at an earlier point in the provider network, eventually the PIM request finds its way to the BMAF. It has been suggested that the border gateway in which the BMAF resides can be made a PIM-SM rendezvous point (RP) to ensure that requests for new groups reach it.

4. Remapping the <Source, Group> Address Pair At the BMAF

The BMAF needs to map from the IPvy <Source, Group> address pair it received back to the corresponding IPvx address pair before propagating the PIM request into the IPvx network. It sends a request to the mapping function to provide that mapping. The mapping function already has this mapping, as a result of the original HMAF request, and returns it to the BMAF. [Editor's note: protocol again to be specified later. It can probably be the same as the one used by the HMAF. Have to work out the security considerations.]

5. Propagation Of the PIM Request Into the IPvx Network

The BMAF propagates translates the PIM request from IPvy to IPvx using the mapping it received. It propagates the request into the IPvx network to complete the construction of the path for the requested multicast stream. If path construction fails, the BMAF SHOULD notify the mapping function so it can mark the IPvx address

pair as bad (so it doesn't get remapped) while releasing the assigned IPvy addresses.

6. Transport of Multicast Media and Unicast RTCP Feedback

If the BMAF receives a multicast packet from the IPvx network, it checks its cache of mappings to locate the IPvy source and group addresses corresponding to the incoming IPvx packet header. It encapsulates the packet in an IPvy header containing the mapped IPvy source and with the destination set to the mapped IPvy group address. It then forwards the packet to the next hop in the multicast tree for that source and group.

When the HMAF receives a multicast packet from the IPvy network, it decapsulates it and forwards it to the host.

When the IPvx host sends unicast RTCP [RFC3550] feedback toward the source, the packets are handled like any other unicast packets. That is, they are processed by the unicast adaptation functions rather than the HMAF and BMAF.

Finally, if the IPvx Host emits multicast packets destined for an any-source multicast group, the processing of the packet is as just described, but with the roles of the HMAF and BMAF reversed.

4. Acknowledgements

This draft started out as draft-tsou-softwire-6rd-multicast-00. Thanks to Joel Halpern for suggesting that it be written as a more general document, since it did not really depend on 6rd. Thanks to Yiu Lee for further comments, which have been used to improve the document.

5. Mapping Request Protocol

To come.

6. Operational Considerations

The proposal presented here incurs the operational expense of provisioning the multicast and unicast address pools at the mapping function. Proper functioning of the system requires that the operator estimate the total number of different IPvx multicast groups and, for source-specific multicast, the total number of individual IPvx sources it wishes to enable simultaneously.

7. IANA Considerations

This memo currently includes no request to IANA.

8. Security Considerations

To come.

9. References

9.1. Normative References

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

[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.

[RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, January 2005.

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

9.2. Informative References

[ID.behave-translated-multicast] Tsou, T., Taylor, T., Zhou, C., and H. Ji, "A Generic Approach to Multicast Translation In Support of IPv6 Transition", January 2011.

[ID.port-control-protocol] Wing, D., "Port Control Protocol (PCP)", January 2011.

[ID.softwire-dslite-multicast] Wang, Q., Qin, J., Sun, P., Boucadair, M., Jacquenet, C., and Y. Lee, "Multicast Extensions to DS-Lite Technique in Broadband Deployments", January 2011.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.

Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

Authors' Addresses

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

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

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

Phone:
Email: cathyzhou@huawei.com

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

Phone:
Email: jihui@chinatelecom.com.cn

Network Working Group
Internet-Draft
Expires: January 10, 2012

M. Xu
Y. Cui
S. Yang
Tsinghua University
C. Metz
G. Shepherd
Cisco Systems

July 9, 2011

Softwire Mesh Multicast
draft-xu-softwire-mesh-multicast-02

Abstract

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

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

Status of this Memo

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

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

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

This Internet-Draft will expire on January 10, 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.

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

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

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

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

One method to accomplish this is to re-use the multicast VPN approach outlined in [10]. MVPN-like schemes can support the softwire mesh scenario and achieve a "many-to-one" mapping between the E-IP client multicast trees and transit core multicast trees. The advantage of this approach is that the number of trees in the I-IP backbone network scales less than linearly with the number of E-IP client trees. Corporate enterprise networks and by extension multicast VPNs have been known to run applications that create a large amount of (S,G) states. Aggregation at the edge contains the (S,G) states that need to be maintained by the network operator supporting the customer VPNs. The disadvantage of this approach is possible inefficient bandwidth and resource utilization if multicast packets are delivered to a receiver AFBR with no attached E-IP receiver.

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

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

2. Terminology

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

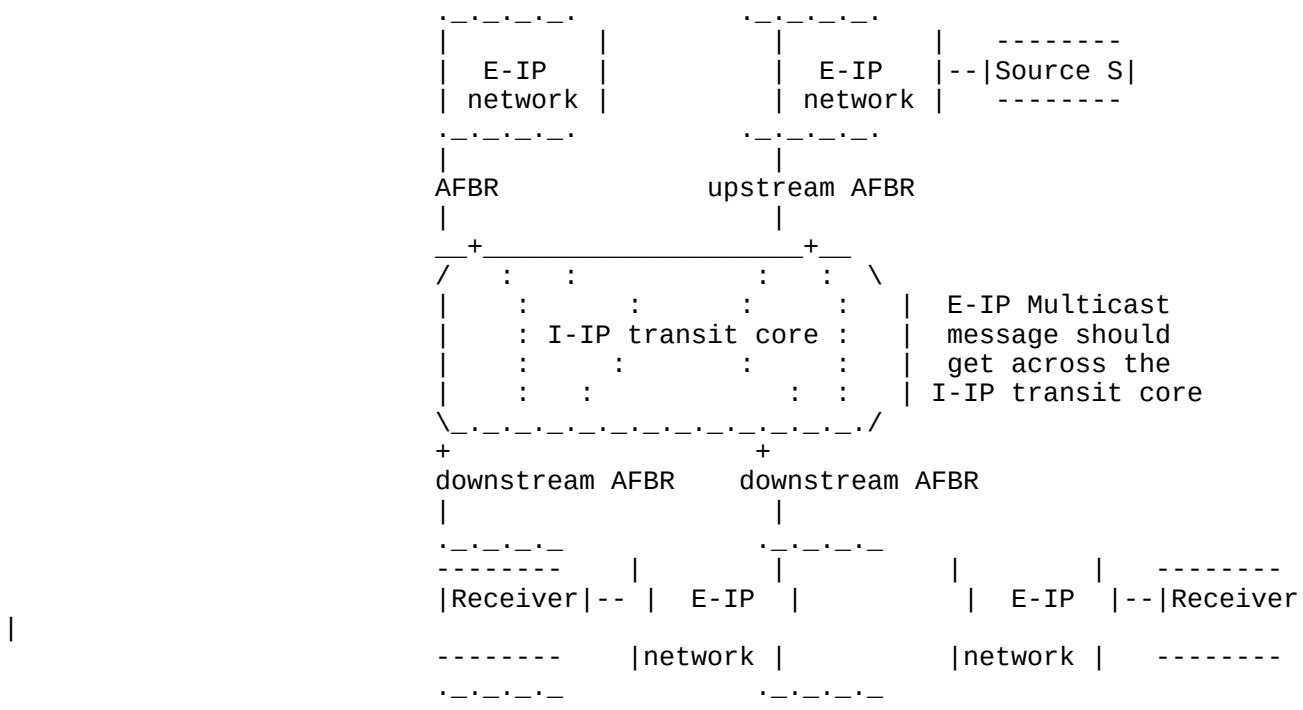


Figure 1: Softwire Mesh Multicast Framework

Terminology used in this document:

o Address Family Border Router (AFBR) - A dual-stack router

interconnecting two or more networks using different IP address families. In the context of softwire mesh multicast, the AFBR runs E-IP and I-IP control planes to maintain E-IP and I-IP multicast states respectively and performs the appropriate encapsulation/ decapsulation of client E-IP multicast packets for transport across the I-IP core. An AFBR will act as a source and/or receiver in an I-IP multicast tree.

o Upstream AFBR: The AFBR router that is located at the upstream of a multicast data flow.

o Downstream AFBR: The AFBR router that is located at the downstream of a multicast data flow.

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

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

o I-IP core tree. A single-source or multi-source distribution tree rooted at one or more AFBR source nodes and branched out to one or more AFBR leaf nodes. An I-IP core Tree is built using standard IP or MPLS multicast signaling protocols operating exclusively inside the I-IP core network. An I-IP core Tree is used to tunnel E-IP multicast packets belonging to E-IP trees across the I-IP core. Another name for an I-IP core Tree is multicast or multipoint softwire.

o E-IP client tree. A single-source or multi-source distribution tree rooted at one or more hosts or routers located inside a client E-IP network and branched out to one or more leaf nodes located in the same or different client E-IP networks.

3. Scenarios of Interest

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

3.1. IPv4-over-IPv6

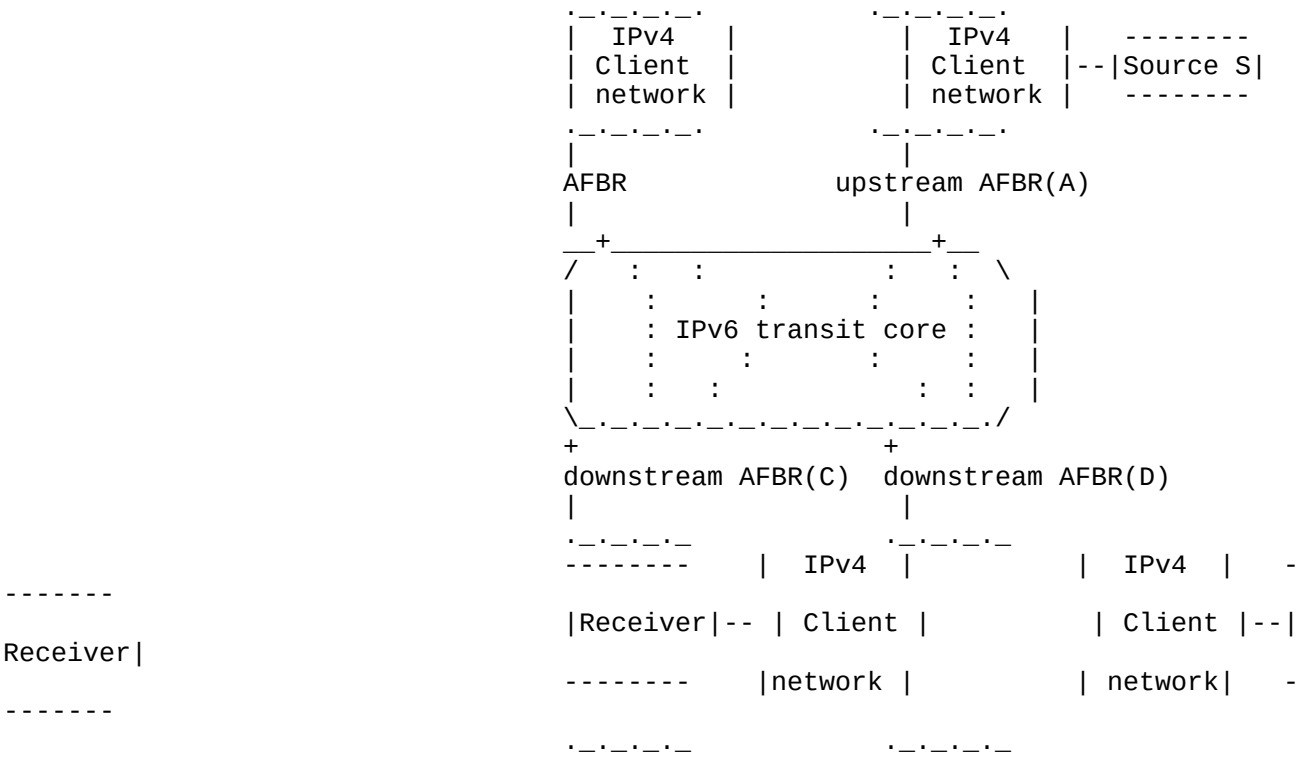


Figure 2: IPv4-over-IPv6 Scenario

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

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

The IPv4-over-IPv6 scenario is an emerging requirement as network operators build out native IPv6 backbone networks. These networks

naturally support native IPv6 services and applications but it is with near 100% certainty that legacy IPv4 networks handling unicast and multicast will need to be accommodated.

3.2. IPv6-over-IPv4

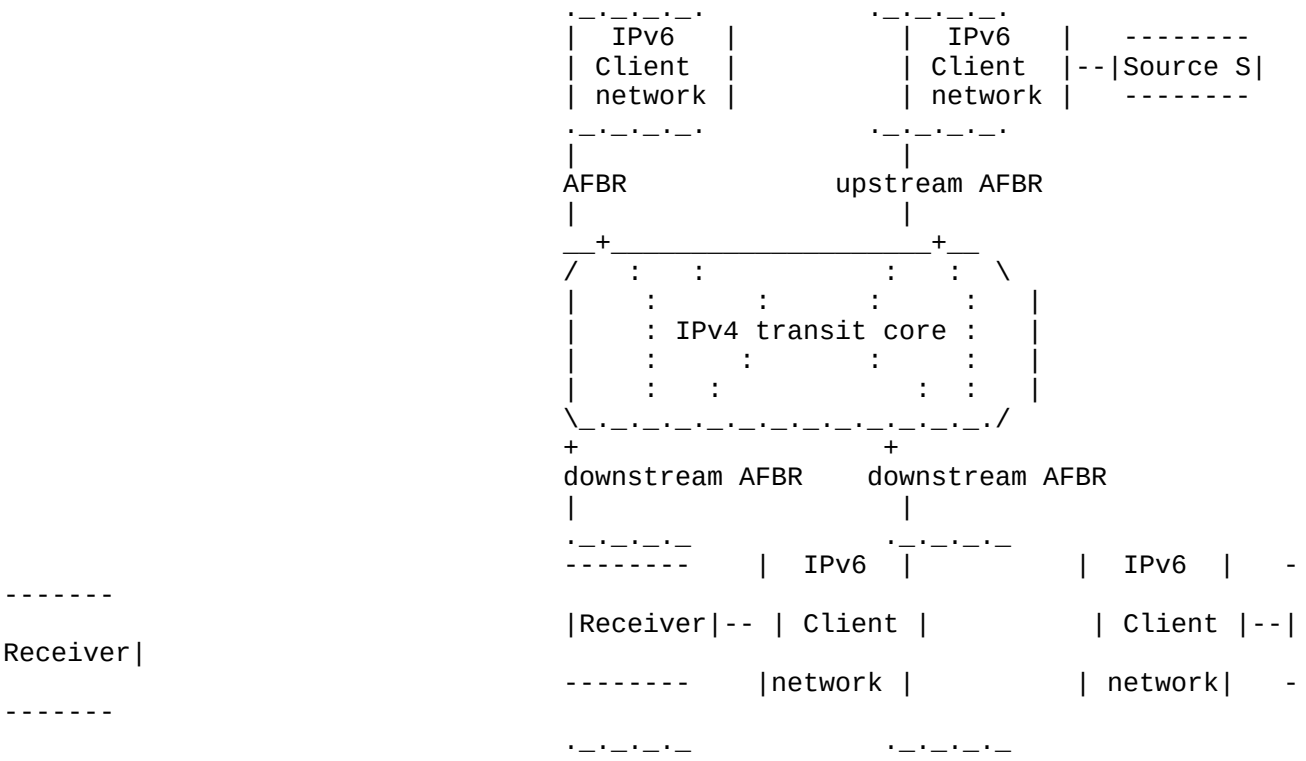


Figure 3: IPv6-over-IPv4 Scenario

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

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

and IPv4 group addresses unless the IPv6 group address is scoped.

As mentioned earlier this scenario is common in the MVPN environment. As native IPv6 deployments and multicast applications emerge from the outer reaches of the greater public IPv4 Internet, it is envisaged that the IPv6 over IPv4 softwire mesh multicast scenario will be a necessary feature supported by network operators.

4. IPv4-over-IPv6

4.1. Mechanism

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

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

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

4.2. Source Address Mapping

There are two kinds of multicast --- ASM and SSM. It’s possible for I-IP network and E-IP network to support different kinds of multicast, and the source address translation rules may vary a lot. There are four scenarios to be discussed in detail:

- E-IP network supports SSM, I-IP network supports SSM
  One possible way to make sure that the translated I-IPv6 PIM message reaches upstream AFBR is to set S’ to a virtual IPv6 address that leads to the upstream AFBR. Figure 4 is the recommended address format based on [9]:

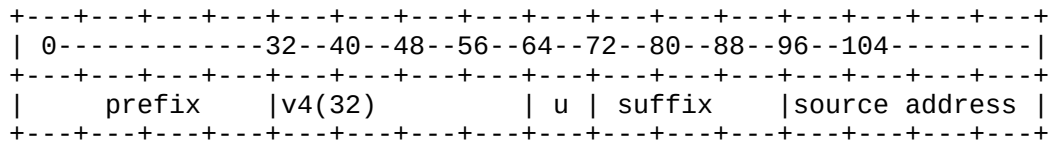


Figure 4: IPv4-Embedded IPv6 Virtual Source Address Format

In this address format, the "prefix" field contains a "Well-Known" prefix or a ISP-defined prefix. An existing "Well-Known" prefix is 64:ff9b, which is defined in [9]; "v4" field is the IP address of one of upstream AFBR's E-IPv4 interface; "u" field is defined in [4], and MUST be set to zero; "suffix" field is reserved for future extensions and SHOULD be set to zero; "source address" field stores the original S.
To make it feasible, the /32 prefix must be known to every AFBR, and AFBRs should not only announce the /96 prefixes of S' to the I-IPv6 network, but also announce the IP addresses of upstream AFBRs' E-IPv4 interface presented in the "v4" field to other AFBRs by MPBGP. In this way, when a downstream AFBR receives a (S,G) message, it can translate it into (S',G') by looking up the IP address of the corresponding AFBR's E-IPv4 interface. Since S' is globally unique and the /96 prefix of S' is known to every router in I-IPv6 network, the translated message will eventually arrive at the corresponding upstream AFBR, and the upstream AFBR can translate the message back to (S,G).

- o E-IP network supports SSM, I-IP network supports ASM
  Since any network that supports ASM should also support SSM, we can construct a SSM tree in I-IP network. The operation in this scenario is the same as that in the first scenario.

- o E-IP network supports ASM, I-IP network supports SSM
  ASM and SSM have the same PIM message format. The main differences between ASM and SSM are RP and (*,G) messages. To make this scenario feasible, we must be able to translate (*,G) messages into (S',G') messages at downstream AFBRs, and translate it back at upstream AFBRs. Assume RP' is the upstream AFBR that locates between RP and the downstream AFBR. When downstream AFBR receives an E-IPv4 PIM (*,G) message, S' can be generated according to the format specified in Figure 4, with "v4" field setting to the IP address of one of RP's E-IPv4 interface and "source address" field setting to *(the IPv4 address of RP). The translated message will eventually arrive at RP'. RP' checks the "source address" field and find the IPv4 address of RP, so RP' judges that this is originally a (*,G) message, then it translates the message back to (*,G) message and forward it to RP.
  Traveling all the way from sources to the RP, and then back down the shared tree may result in the multicast data packets passing through RP' twice, which brings about undesirable increased latency or bandwidth consumption. For this reason, RP' MAY perform a "cut-through", namely when RP' receives multicast data packets sent from sources to RP, it not only forwards them to RP, but also forwards them directly onto the multicast tree built in the I-IPv6 network. (S,G,rpt) messages should be sent towards RP to avoid reduplication.

- o E-IP network supports ASM, I-IP network supports ASM
  To keep it as simple as possible, we treat I-IP network as SSM and the solution is the same as the third scenario.

4.3. Group Address Mapping

For IPv4-over-IPv6 scenario, a simple algorithmic mapping between IPv4 multicast group addresses and IPv6 group addresses is supported. [11] has already defined an applicable format. Figure 5 is a reminder of the format:

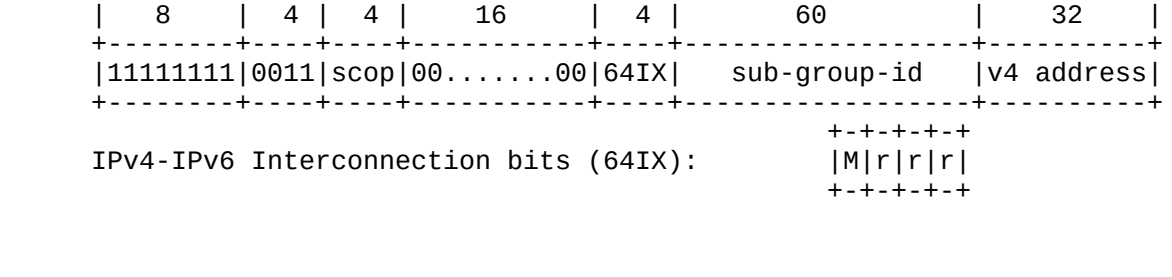


Figure 5: IPv4-Embedded IPv6 Multicast Address Format: SSM Mode

The high order bits of the I-IPv6 address range will be fixed for mapping purposes. With this scheme, each IPv4 multicast address can be mapped into an IPv6 multicast address(with the assigned prefix), and each IPv6 multicast address with the assigned prefix can be mapped into IPv4 multicast address.

4.4. Actions performed by AFBR

The following actions are performed by AFBRs:

- o Receive E-IPv4 PIM messages
  When a downstream AFBR receives an E-IPv4 PIM message, it should check the address family of the next-hop towards the destination. If the address family is IPv4, the AFBR should forward the message without any translation; otherwise it should take the following operation.

- o Translate E-IPv4 PIM messages into I-IPv6 PIM messages
  E-IPv4 PIM message with S(or *) and G is translated into I-IPv6 PIM message with S' and G' following the rules specified above.

- o Transmit I-IPv6 PIM messages
  The downstream AFBR sends the I-IPv6 PIM message to the upstream AFBR. When the upstream AFBR receives this I-IPv6 PIM message, it

checks the prefix of the source address and judges that the message is a translated message, then translates the message back to E-IPv4 PIM message and sends it towards source or RP.

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

5. IPv6-over-IPv4

5.1. Mechanism

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

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

5.2. Source Address Mapping

There are two kinds of multicast --- ASM and SSM. It’s possible for I-IP network and E-IP network to support different kind of multicast, and the source address translation rules may vary a lot. There are four scenarios to be discussed in detail:

- E-IP network supports SSM, I-IP network supports SSM
  To make sure that the translated I-IPv4 PIM message reaches the upstream AFBR, we need to set S’ to an IPv4 address that leads to the upstream AFBR. But due to the non-"one-to-one" mapping of E-IPv6 to I-IPv4 unicast address, the upstream AFBR is unable to remap the I-IPv4 source address to the original E-IPv6 source address without any constraints.
  We apply a fixed IPv6 prefix and static mapping to solve this problem. A recommended source address format is defined in [9]. Figure 6 is a reminder of the format:

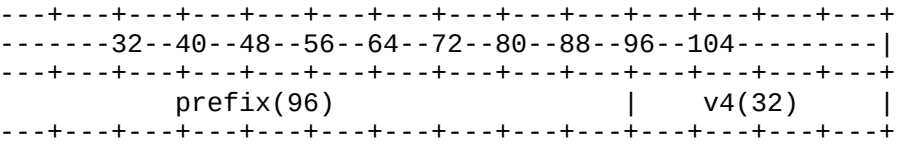


Figure 6: IPv4-Embedded IPv6 Source Address Format

In this address format, the "prefix" field contains a "Well-Known" prefix or a ISP-defined prefix. An existing "Well-Known" prefix is 64:ff9b, which is defined in [9]; "v4" field is the corresponding I-IPv4 source address.
To make it feasible, the /96 prefix must be known to every AFBR, every E-IPv6 address of sources that support mesh multicast MUST follow the format specified in Figure 6, and the corresponding upstream AFBR should announce the I-IPv4 address in "v4" field to the I-IPv4 network. In this way, when a downstream AFBR receives a (S,G) message, it can translate it into (S',G') by simply take off the prefix in S. Since S' is known to every router in I-IPv4 network, the translated message will eventually arrive at the corresponding upstream AFBR, and the upstream AFBR can translate the message back to (S,G) by appending the prefix to S'.

- o E-IP network supports SSM, I-IP network supports ASM
  Since any network that supports ASM should also support SSM, we can construct a SSM tree in I-IP network. The operation in this scenario is the same as that in the first scenario.

- o E-IP network supports ASM, I-IP network supports SSM
  ASM and SSM have the same PIM message format. The main differences between ASM and SSM are RP and (*,G) messages. To make this scenario feasible, we must be able to translate (*,G) messages into (S',G') messages at downstream AFBRs and translate it back at upstream AFBRs. Here, the E-IPv6 address of RP MUST follow the format specified in Figure 6. Assume RP' is the upstream AFBR that locates between RP and the downstream AFBR. When a downstream AFBR receives a (*,G) message, it can translate it into (S',G') by simply take off the prefix in *(the E-IPv6 address of RP). Since S' is known to every router in I-IPv4 network, the translated message will eventually arrive at RP'. RP' knows that S' is the mapped I-IPv4 address of RP, so RP' will translate the message back to (*,G) by appending the prefix to S' and forward it to RP.
  Traveling all the way from sources to the RP, and then back down the shared tree may result in the multicast data packets passing through RP' twice, which brings about undesirable increased latency or bandwidth consumption. For this reason, RP' MAY perform a "cut-through", namely when RP' receives multicast data packets sent from sources to RP, it not only forwards them to RP, but also forwards them directly onto the multicast tree built in the I-IPv6 network. (S,G,rpt) messages should be sent towards RP to avoid reduplication.

- o E-IP network supports ASM, I-IP network supports ASM
  To keep it as simple as possible, we treat I-IP network as SSM and the solution is the same as the third scenario.

5.3. Group Address Mapping

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

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

5.4. Actions performed by AFBR

The following actions are performed by AFBRs

- Receive E-IPv6 PIM messages
  When a downstream AFBR receives an E-IPv6 PIM message, it should check the address family of the upstream router. If the address family is IPv6, the AFBR should not translate this message; otherwise it should take the following operation.

- Translate E-IPv6 PIM messages into I-IPv4 PIM messages
  E-IPv6 PIM message with S(or *) and G is translated into I-IPv4 PIM message with S’ and G’ following the rules specified above.

- Transmit I-IPv4 PIM messages
  The downstream AFBR sends the I-IPv4 PIM message to the upstream AFBR. When the upstream AFBR receives this I-IPv4 PIM message, it checks the source address and judges that the message is a translated message, then translates the message back to E-IPv6 PIM message and sends it towards source or RP.

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

6. Security Considerations

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

7. IANA Considerations

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

8. References

8.1. Normative References

[1] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[2] Foster, B. and F. Andreasen, "Media Gateway Control Protocol (MGCP) Redirect and Reset Package", RFC 3991, February 2005.

[3] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

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

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

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

[7] Wijnands, IJ., Boers, A., and E. Rosen, "The Reverse Path Forwarding (RPF) Vector TLV", RFC 5496, March 2009.

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

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

8.2. Informative References

[10] Aggarwal, R., Bandi, S., Cai, Y., Morin, T., Rekhter, Y., Rosen, E., Wijnands, I., and S. Yasukawa, "Multicast in MPLS/ BGP IP VPNs", draft-ietf-l3vpn-2547bis-mcast-10 (work in progress), January 2010.

[11] Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X., and M. Xu, "IPv4-Embedded IPv6 Multicast Address Format", draft-boucadair-behave-64-multicast-address-format-02 (work in progress), June 2011.

Appendix A. Acknowledgements

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

Authors' Addresses

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

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

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

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

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

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

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

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

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

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