Internet Engineering Task Force	A. Durand, Ed.	
Internet-Draft	Comcast	
Intended status: Standards Track	March 08, 2010	
Expires: September 9, 2010		

TOC

Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion
draft-ietf-softwire-dual-stack-lite-04

Abstract

This document revisits the dual-stack model and introduces the dual-stack lite technology aimed at better aligning the costs and benefits of deploying IPv6. Dual-stack lite enables a broadband service provider to share IPv4 addresses among customers by combining two well-known technologies: IP in IP (IPv4-in-IPv6) and Network Address Translation (NAT).

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.
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.”
The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.
This Internet-Draft will expire on September 9, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.
This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted

from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

Table of Contents

1. Introduction
2. Requirements language
3. Terminology
4. Deployment scenarios
    4.1. Access model
    4.2. Home gateway
    4.3. Directly connected device
5. B4 element
    5.1. Definition
    5.2. Encapsulation
    5.3. Fragmentation and Reassembly
    5.4. AFTR discovery
    5.5. DNS
    5.6. Interface initialization
    5.7. Well-known IPv4 address
6. AFTR element
    6.1. Definition
    6.2. Encapsulation
    6.3. Fragmentation and Reassembly
    6.4. DNS
    6.5. Well-known IPv4 address
    6.6. Extended binding table
7. Network Considerations
    7.1. Tunneling
    7.2. VPN
    7.3. Multicast considerations
8. NAT considerations
    8.1. NAT pool
    8.2. NAT conformance
    8.3. Application Level Gateways (ALG)
    8.4. Port allocation
        8.4.1. How many ports per customers?
        8.4.2. Dynamic port assignment considerations
        8.4.3. Subscriber controlled port assignment
    8.5. Other considerations about sharing global IPv4 addresses
9. Acknowledgements
10. IANA Considerations
11. Security Considerations
12. Author's Addresses
13. Appendix A: future DS-Lite extensions
    13.1. Static port reservation

13.1.1. Port forwarding model
13.1.2. A+P model
13.2. Dynamic port reservation
13.2.1. UPnP
13.2.2. NAT-PMP
13.2.3. DHCPv6
14. Appendix B: Examples
14.1. Gateway based architecture
14.1.1. Example message flow
14.1.2. Translation details
14.2. Host based architecture
14.2.1. Example message flow
14.2.2. Translation details
15. Appendix C: Deployment considerations
15.1. AFTR service distribution and horizontal scaling
15.2. Horizontal scaling
15.3. High availability
15.4. Logging
16. References
16.1. Normative references
16.2. Informative references
§ Author's Address

---

1. Introduction

TOC

The common thinking for more than 10 years has been that the transition to IPv6 will be based on the dual stack model and that most things would be converted this way before we ran out of IPv4.

However, this has not happened. The IANA free pool of IPv4 addresses will be depleted soon, well before any significant IPv6 deployment will have occurred.

This document revisits the dual-stack model and introduces the dual-stack lite technology aimed at better aligning the costs and benefits of deploying IPv6. Dual-stack lite will provide the necessary bridge between the two protocols, offering an evolution path of the Internet post IANA IPv4 depletion.

Dual-stack lite enables a broadband service provider to share IPv4 addresses among customers by combining two well-known technologies: IP in IP (IPv4-in-IPv6) and NAT.

This document makes a distinction between a dual-stack capable and a dual-stack provisioned device. The former is a device that has code that implements both IPv4 and IPv6, from the network layer to the applications. The latter is a similar device that has been provisioned with both an IPv4 and an IPv6 address on its interface(s). This document will also further refine this notion by distinguishing between

interfaces provisioned directly by the service provider from those provisioned by the customer.
Pure IPv6-only devices (i.e. devices that do not include an IPv4 stack) are outside of the scope of this document.

---

2. Requirements language

TOC

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 (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.) [RFC2119].

---

3. Terminology

TOC

The technology described in this document is known as dual-stack lite. The abbreviation DS-Lite will be used along this text.
This document also introduces two new terms: the DS-Lite Basic Bridging BroadBand element (B4) and the DS-Lite Address Family Transition Router element (AFTR).

---

4. Deployment scenarios

TOC

4.1. Access model

TOC

Instead of relying on a cascade of NATs, the dual-stack lite model is built on IPv4-in-IPv6 tunnels to cross the network to reach a carrier-grade IPv4-IPv4 NAT (the AFTR) where customers will share IPv4 addresses. There are numbers of benefits to this approach:

* This technology decouples the deployment of IPv6 in the service provider network (up to the customer premise equipment or CPE) from the deployment of IPv6 in the global Internet and in customer applications & devices.

* The management of the service provider access networks is simplified by leveraging the large IPv6 address space.

Overlapping private IPv4 address spaces are not required to support very large customer bases.

*As tunnels can terminate anywhere in the service provider network, this architecture leads itself to horizontal scaling and provides great flexibility to adapt to changing traffic load.

*Tunnels provide a direct connection between B4 and the AFTR. This can be leveraged to enable customers and their applications to control how the NAT function of the AFTR is performed.

A key characteristic of this approach is that communications between end-nodes stay within their address family. IPv6 sources only communicate with IPv6 destinations, IPv4 sources only communicate with IPv4 destinations. There is no protocol family translation involved in this approach. This simplifies greatly the task of applications that may carry literal IP addresses in their payload.

4.2. Home gateway

TOC

This section describes home Local Area networks characterized by the presence of a home gateway provisioned only with IPv6 by the service provider.
A DS-Lite home gateway is an IPv6 aware home gateway with a B4 Interface implemented in the WAN interface.
A DS-Lite home gateway SHOULD NOT operate a NAT function on a B4 interface, as the NAT function will be performed by the AFTR in the service provider's network. That will avoid accidentally operating in a double NAT environment.
However, it SHOULD operate its own DHCP(v4) server handing out [RFC1918] (Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, “Address Allocation for Private Internets,” February 1996.) address space (e.g. 192.168.0.0/16) to hosts in the home. It SHOULD advertise itself as the default IPv4 router to those home hosts. It SHOULD also advertise itself as a DNS server in the DHCP Option 6 (DNS Server). Additionally, it SHOULD operate a DNS proxy to accept DNS IPv4 requests from home hosts and send them using IPv6 to the service provider DNS servers, as described in Section 5.5.
Note: if an IPv4 home host decides to use another IPv4 DNS server, the DS-Lite home gateway will forward those DNS requests via the B4 interface, the same way it forwards any regular IPv4 packets.
IPv6 capable devices directly reach the IPv6 Internet. Packets simply follow IPv6 routing, they do not go through the tunnel, and are not subject to any translation. It is expected that most IPv6 capable devices will also be IPv4 capable and will simply be configured with an IPv4 RFC1918 style address within the home network and access the IPv4 Internet the same way as the legacy IPv4-only devices within the home.

Pure IPv6-only devices (i.e. devices that do not include an IPv4 stack) are outside of the scope of this document.

4.3. Directly connected device

TOC

In broadband home networks, sometime devices are directly connected to the broadband service provider. They are connected straight to a modem, without a home gateway.
Under this scenario, the customer device is a dual-stack capable host that is only provisioned by the service provider only with IPv6. The device itself acts as a B4 element and the IPv4 service is provided by an IPv4-in-IPv6 tunnel, just as in the home gateway case. That device can run any combinations of IPv4 and/or IPv6 applications.
A directly connected DS-Lite device SHOULD send its DNS requests over IPv6 to the IPv6 DNS server it has been configured to use.
Similarly to the previous sections, IPv6 packets follow IPv6 routing, they do not go through the tunnel, and are not subject to any translation.
The support of IPv4-only devices and IPv6-only devices in this scenario is out of scope for this document.

---

5. B4 element

TOC

5.1. Definition

TOC

The B4 element is a function implemented on a dual-stack capable node, either a directly connected device or a home gateway, that creates a tunnel to an AFTR.

5.2. Encapsulation

TOC

The tunnel is a multi-point to point IPv4-in-IPv6 tunnel ending on a service provider AFTR.
See section 7.1 for additional tunneling considerations.
Note: at this point, DS-Lite only defines IPv4-in-IPv6 tunnels, however other types of encapsulation could be defined in the future.

5.3. Fragmentation and Reassembly

TOC

Using an encapsulation (IPv4-in-IPv6 or anything else) to carry IPv4 traffic over IPv6 will reduce the effective MTU of the datagram. Unfortunately, path MTU discovery [RFC1191] (Mogul, J. and S. Deering, “Path MTU discovery,” November 1990.) is not a reliable method to deal with this problem.

A solution to deal with this problem is for the service provider to increase the MTU size of all the links between the B4 element and the AFTR elements by at least 40 bytes to accommodate both the IPv6 encapsulation header and the IPv4 datagram without fragmenting the IPv6 packet.

However, as not all service providers will be able to increase their link MTU, the B4 element MUST perform fragmentation and reassembly if the outgoing link MTU cannot accommodate for the extra IPv6 header. Fragmentation MUST happen after the encapsulation on the IPv6 packet. Reassembly MUST happen before the decapsulation of the IPv6 header. Detailed procedure has been specified in [RFC2473] (Conta, A. and S. Deering, “Generic Packet Tunneling in IPv6 Specification,” December 1998.) Section 7.2.

5.4. AFTR discovery

TOC

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] (Hankins, D. and T. Mrugalski, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Options for Dual- Stack Lite,” March 2010.).

5.5. DNS

TOC

A B4 element is only configured from the service provider with IPv6. As such, it can only learn the address of a DNS recursive server through DHCPv6 (or other similar method over IPv6). As DHCPv6 only defines an option to get the IPv6 address of such a DNS recursive server, the B4 element cannot easily discover the IPv4 address of such a recursive DNS server, and as such will have to perform all DNS resolution over IPv6. The B4 element can pass this IPv6 address to downstream IPv6 nodes, but not to downstream IPv4 nodes. As such, the B4 element SHOULD implement

a DNS proxy, following the recommendations of [RFC5625] (Bellis, R., “DNS Proxy Implementation Guidelines,” August 2009.).

5.6. Interface initialization

TOC

Initialization of the interface including a B4 element is out-of-scope in this specification.

5.7. Well-known IPv4 address

TOC

Any locally unique IPv4 address could be configured on the IPv4-in-IPv6 tunnel to represent the B4 element. Configuring such an address is often necessary when the B4 element is sourcing IPv4 datagrams directly over the tunnel. In order to avoid conflicts with any other address, IANA has defined a well-known range, 192.0.0.0/29.
192.0.0.0 is the reserved subnet address. 192.0.0.1 is reserved for the AFTR element. The B4 element MAY use any other addresses within the 192.0.0.0/29 range.
Note: a range of addresses has been reserved for this purpose. The intent is to accommodate nodes implementing multiple B4 elements.

---

6. AFTR element

TOC

6.1. Definition

TOC

An AFTR element is the combination of an IPv4-in-IPv6 tunnel end-point and an IPv4-IPv4 NAT implemented on the same node.

6.2. Encapsulation

TOC

The tunnel is a point-to-multipoint IPv4-in-IPv6 tunnel ending at the B4 elements.
See section 7.1 for additional tunneling considerations.
Note: at this point, DS-Lite only defines IPv4-in-IPv6 tunnels, however other types of encapsulation could be defined in the future.

6.3. Fragmentation and Reassembly

TOC

As noted previously, fragmentation and reassembly need to be taken care of by the tunnel end-points. As such, the AFTR MUST perform fragmentation and reassembly if the underlying link MTU cannot accommodate the extra IPv6 header of the tunnel. Fragmentation MUST happen after the encapsulation on the IPv6 packet. Reassembly MUST happen before the decapsulation of the IPv6 header. Detailed procedure has been specified in [RFC2473] (Conta, A. and S. Deering, “Generic Packet Tunneling in IPv6 Specification,” December 1998.) Section 7.2. Fragmentation at the Tunnel Entry-Point is a light-weight operation. In contrast, reassembly at the Tunnel Exit-Point can be expensive. When the Tunnel Exit-Point receives the first fragmented packet, it must wait for the second fragmented packet to arrive in order to reassemble the two fragmented IPv6 packets for decapsulation. This requires the Tunnel Exit-Point to buffer and keep track of fragmented packets. Consider that the AFTR is the Tunnel Exit-Point for many tunnels. If many clients simultaneously source large number of fragmented packets to the AFTR, this will require the AFTR to buffer and consume enormous resources to keep track of the flows. This reassembly process will significantly impact the AFTR performance. However, this impact only happens when many clients simultaneously source large IPv4 packets. Since we believe that majority of the clients will receive large IPv4 packets (such as watching video streams) instead of sourcing large IPv4 packets (such as sourcing video streams), so reassembly is only a fraction of the overall AFTR's workload.
Methods to avoid fragmentation, such as rewriting the TCP MSS option or using technologies such as Subnetwork Encapsulation and Adaptation Layer defined in [I-D.templin-seal] (Templin, F., “The Subnetwork Encapsulation and Adaptation Layer (SEAL),” August 2008.) are out of scope for this document.

6.4. DNS

TOC

As noted previously, DS-Lite node implementing a B4 elements will perform DNS resolution over IPv6. As such, very few, if any, DNS packets will flow through the AFTR element.

TOC

6.5. Well-known IPv4 address

The AFTR MAY use the well-known IPv4 address 192.0.0.1 reserved by IANA to configure the IPv4-in-IPv6 tunnel. That address can then be used to report ICMP problems and will appear in traceroute outputs.

6.6. Extended binding table

TOC

The NAT binding table of the AFTR element is extended to include the source IPv6 address of the incoming packets. This IPv6 address is used to disambiguate between the overlapping IPv4 address space of the service provider customers.
By doing a reverse look-up in the extended IPv4 NAT binding table, the AFTR knows how to reconstruct the IPv6 encapsulation when the packets comes back from the Internet. That way, there is no need to keep a static configuration for each tunnel.

---

7. Network Considerations

TOC

7.1. Tunneling

TOC

Tunneling MUST be done in accordance to [RFC2473] (Conta, A. and S. Deering, “Generic Packet Tunneling in IPv6 Specification,” December 1998.) and [RFC4213] (Nordmark, E. and R. Gilligan, “Basic Transition Mechanisms for IPv6 Hosts and Routers,” October 2005.). Traffic classes ([RFC2474] (Nichols, K., Blake, S., Baker, F., and D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,” December 1998.)) from the IPv4 headers SHOULD be carried over to the IPv6 headers and vice versa.

7.2. VPN

TOC

Dual-stack lite implementations SHOULD not interfere with the functioning of IPv4 or IPv6 VPNs.

TOC

7.3. Multicast considerations

Multicast is out-of-scope in this document.

---

8. NAT considerations

TOC

8.1. NAT pool

TOC

AFTRs MAY operate distinct, non overlapping NAT pools. Those NAT pools do not have to be continuous.

8.2. NAT conformance

TOC

A dual-stack lite AFTR SHOULD implement behavior conforming to the best current practice, currently documented in [RFC4787] (Audet, F. and C. Jennings, “Network Address Translation (NAT) Behavioral Requirements for Unicast UDP,” January 2007.), [RFC5382] (Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, “NAT Behavioral Requirements for TCP,” October 2008.) and [RFC5508] (Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, “NAT Behavioral Requirements for ICMP,” April 2009.). Other requirements for AFTRs can be found in [I-D.nishitani-cgn] (Yamagata, I., Nishitani, T., Miyakawa, S., Nakagawa, A., and H. Ashida, “Common requirements for IP address sharing schemes,” March 2010.).

8.3. Application Level Gateways (ALG)

TOC

The AFTR should only perform a minimum number of ALG for the classic applications such as FTP, RTSP/RTP, IPsec and PPTP VPN pass-through and enable the users to use their own ALG on statically or dynamically reserved ports instead.

8.4. Port allocation

TOC

8.4.1. How many ports per customers?

TOC

Because IPv4 addresses will be shared among customers and potentially a large address space reduction factor may be applied, in average, only a limited number N of TCP or UDP port numbers will be available per customer. This means that applications opening a very large number of TCP ports may have a harder time to work. For example, it has been reported that a very well know web site was using AJAX techniques and was opening up to 69 TCP ports per web page. If we make the hypothesis of an address space reduction of a factor 100 (one IPv4 address per 100 customers), and 65k ports per IPv4 addresses available, that makes an average of N = 650 ports available simultaneously to be shared among the various devices behind the dual-stack lite tunnel end-point.

There is an important operational difference if those N ports are pre-allocated in a cookie-cutter fashion versus allocated on demand by incoming connections. This is a difference between an average of N ports and a maximum of N ports. Several service providers have reported an average number of connections per customer in the single digits. At the opposite end, thousands or tens of thousands of ports could be use in a peak by any single customer browsing a number of AJAX/Web 2.0 sites.

As such, service providers allocating a fixed number of ports per user should dimension the system with a minimum of N = several thousands of ports for every user. This would bring the address space reduction ratio to a single digit. Service providers using a smaller number of ports per user (N in the hundreds) should expect customers applications to break in a more or less random way over time.

In order to achieve higher address space reduction ratios, it is recommended that service provider do not use this cookie-cutter approach, and, on the contrary, allocate ports as dynamically as possible, just like on a regular NAT. With an average number of connections per customers in the single digit, having an address space reduction of a factor 100 is realistic. However, service providers should exercise caution and make sure their pool of port numbers does not go too low. The actual maximum address space reduction factor is unknown at this time.

8.4.2. Dynamic port assignment considerations

TOC

When dynamic port assignment is used to maximize the number of subscribers sharing the AFTR global IPv4 addresses, the AFTR should implement checks to avoid DOS attack through exhaustion of available ports. It should also avoid mapping any one subscriber's "flows" across more than one global IPv4 address.

8.4.3. Subscriber controlled port assignment

TOC

Dynamic port assignment precludes inbound access to subscriber servers, just as in a home gateway NAT. Inbound access to subscriber servers can be provided through pre-assigned and/or reserved port mappings in the AFTR. Specifying the mechanisms for managing and signaling these reserved port mappings is out of scope for this document, however some techniques are mentioned in appendix A as examples.

8.5. Other considerations about sharing global IPv4 addresses

TOC

More considerations on sharing the port space of IPv4 addresses can be found in [I-D.ford-shared-addressing-issues] (Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, “Issues with IP Address Sharing,” March 2010.).

9. Acknowledgements

TOC

The authors would like to acknowledge the role of Mark Townsley for his input on the overall architecture of this technology by pointing this work in the direction of [I-D.droms-softwires-snat] (Droms, R. and B. Haberman, “Softwires Network Address Translation (SNAT),” July 2008.). Note that this document results from a merging of [I-D.durand-dual-stack-lite] (Durand, A., “Dual-stack lite broadband deployments post IPv4 exhaustion,” July 2008.) and [I-D.droms-softwires-snat] (Droms, R. and B. Haberman, “Softwires Network Address Translation (SNAT),” July 2008.).Also to be acknowledged are the many discussions with a number of people including Shin Miyakawa, Katsuyasu Toyama, Akihide Hiura, Takashi Uematsu, Tetsutaro Hara, Yasunori Matsubayashi, Ichiro Mizukoshi. The author would also like to thank David Ward, Jari Arkko, Thomas Narten and Geoff Huston for their constructive feedback. Special thanks go to Dave Thaler and Dan Wing for their reviews and comments.

10. IANA Considerations

TOC

This draft request IANA to allocate a well know IPv4 192.0.0.0/29 network prefix. That range is used to number the dual-stack lite interfaces. Reserving a /29 allows for 6 possible interfaces on a

multi-home node. The IPv4 address 192.0.0.1 is reserved as the IPv4 address of the default router for such dual-stack lite hosts.

---

11. Security Considerations

TOC

Security issues associated with NAT have long been documented. See [RFC2663] (Srisuresh, P. and M. Holdrege, “IP Network Address Translator (NAT) Terminology and Considerations,” August 1999.) and [RFC2993] (Hain, T., “Architectural Implications of NAT,” November 2000.).

However, moving the NAT functionality from the home gateway to the core of the service provider network and sharing IPv4 addresses among customers create additional requirements when logging data for abuse usage. With any architecture where an IPv4 address does not uniquely represent an end host, IPv4 addresses and a timestamps are no longer sufficient to identify a particular broadband customer. Additional information such as transport protocol information will be required for that purpose. For example, we suggest to log the transport port number for TCP and UDP connections.

The AFTR performs translation functions for interior IPv4 hosts at RFC 1918 addresses or at the IANA reserved address range (TBA by IANA). If the interior host is properly using the authorized IPv4 address with the authorized transport protocol port range such as A+P semantic for the tunnel, the AFTR can simply forward without translation to permit the authorized address and port range to function properly. All packets with unauthorized interior IPv4 addresses or with authorized interior IPv4 address but unauthorized port range MUST NOT be forwarded by the AFTR. This prevents rogue devices from launching denial of service attacks using unauthorized public IPv4 addresses in the IPv4 source header field or unauthorized transport port range in the IPv4 transport header field. For example, rogue devices could bombard a public web server by launching TCP SYN ACK attack. The victim will receive TCP SYN from random IPv4 source addresses at a rapid rate and deny TCP services to legitimate users.

With IPv4 addresses shared by multiple users, ports become a critical resource. As such, some mechanisms need to be put in place by an AFTR to limit port usage, either by rate-limiting new connections or putting a hard limit on the maximum number of port usable by single user. If this number is high enough, it should not interfere with normal usage and still provide reasonable protection of the shared pool. More considerations on ports allocation and port exhaustion can be found in section 8.4.

More considerations on sharing IPv4 addresses can be found in "I-D.ford-shared-addressing-issues".

AFTRs should support ways to limit service to registered customers. If strict IPv6 ingress filtering is deployed in the broadband network to

prevent IPv6 address spoofing and dual-stack lite service is restricted to those customers, then tunnels terminating at the AFTR and coming from registered customer IPv6 addresses cannot be spoofed. Thus a simple access control list on the tunnel transport source address is all that is required to accept traffic on the southbound interface of an AFTR.

If IPv6 address spoofing prevention is not in place, the AFTR should perform further sanity checks on the IPv6 address of incoming IPv6 packets. For example, it should check if the address has really been allocated to an authorized customer.

---

12. Author's Addresses

TOC

This document is the result of the work of the following authors:

Alain Durand
Comcast
1, Comcast center
Philadelphia, PA 19103
USA
Email: alain_durand@cable.comcast.com

Ralph Droms
Cisco
1414 Massachusetts Avenue
Boxborough, MA 01714
USA
Phone: +1 978.936.1674
Email: rdroms@cisco.com

Brian Haberman
Johns Hopkins University Applied Physics Lab
11100 Johns Hopkins Road
Laurel, MD 20723-6099
USA
Phone: +1 443 778 1319
Email: brian@innovationslab.net

James Woodyatt
Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
USA
Email: jhw@apple.com

Yiu Lee
Comcast
1, Comcast center
Philadelphia, PA 19103
USA
Email: yiu_lee@cable.comcast.com

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
USA
Phone: +1 206 780 0431 x1
Email: randy@psg.com

---

13. Appendix A: future DS-Lite extensions

TOC

Techniques discussed below are not part of the core dual-stack lite specification and will be developed in separate documents. They are only listed here as examples.
Applications expecting incoming connections, such a peer-to-peer applications, have become popular. Those applications use a very limited number of ports, usually a single one. Making sure those applications keep working in a dual-stack lite environment is important. Similarly, there is a growing list of applications that require some kind of ALG to work through a NAT. Service provider AFTRs should not prevent the deployment of such applications. As such, there is a legitimate need to leave certain ports under the control of the end user. This argues for a hybrid environment, where most ports are dynamically managed by the AFTR in a shared pool and a limited number are dedicated per users and controlled by them.

TOC

13.1. Static port reservation

A service provider can reserve a number of static ports per user. Note: those could be TCP and/or UDP ports. The simplest model to allow users to control the associated NAT bindings is to offer a web interface (for example as part of the service provider portal) where, once authenticated, a user can configure each dedicated external IPv4 address/port binding on the AFTR either using the port forwarding semantic or the A+P semantic.
Note: The exact number of ports reserved per user is left at the discretion of the service provider.

13.1.1. Port forwarding model

TOC

In this model, the subscriber directs the AFTR to rewrite the destination address in those incoming packets to a private IPv4 address within the home network. For obvious security reasons, redirection to global IPv4 address should not be authorized. Note: this behavior is very similar to the port forwarding function found in most home gateways.

13.1.2. A+P model

TOC

The subscriber directs the AFTR to forward incoming traffic on a given address/port to the dual-stack lite home gateway, and let this device deal with it. This required support for A+P (Bush, R., “The A+P Approach to the IPv4 Address Shortage,” October 2009.)
[I-D.ymbk-aplusp] semantic on both the AFTR and on the home gateway. In particular, an A+P aware home router can locally NAT A+P packets to and from internal hosts. Alternatively, it can forward directly the traffic to those hosts if they are configured, for example, with A+P secondary address and ports.
An AFTR forwards packets in the A+P range directly to and from the tunnels without NAT.

13.2. Dynamic port reservation

TOC

TOC

13.2.1. UPnP

A B4 element can act as a UPnP relay, forwarding UPnP messages over the tunnel to the AFTR. This may work in some cases, but not all the time. Some applications insist on running on a well-known port number (or port range) using UPnP to request the NAT to reserve that port. Those ports may or may not be available; they could be used by another customer. Using UPnP, a NAT box does not have any way to redirect such applications to use another port, the only option is to deny the request. Those applications typically then cycle through a small range of ports (typically 10 or so) until they abort. The likelihood of those ports being all already in use by other users is an inverse function of the address space reduction, ie, how many users are sharing the same address.
Note: the UPnP forum has been reported to address this issue in an upcoming version of the IGD profile.

13.2.2. NAT-PMP

TOC

NAT-PMP (Cheshire, S., “NAT Port Mapping Protocol (NAT-PMP),” April 2008.) [I-D.cheshire-nat-pmp] offers a better semantic, by enabling the NAT to redirect the application to use another unallocated port. A B4 element could proxy the NAT-PMP messages to the AFTR through the tunnel.

13.2.3. DHCPv6

TOC

If more ports need to be reserved outside of that static dedicated range, a DHCPv6 option such as [I-D.bajko-v6ops-port-restricted-ipaddr-assign] (Bajko, G. and T. Savolainen, “Port Restricted IP Address Assignment,” November 2008.) may also be an interesting approach. This may be limited to the A+P semantic mentioned above, as there might not be a way to explicitly control the port forwarding semantic. Also, there are concerns that this would lead to a cookie cutter distribution of ports per customers, dramatically reducing the ratio of customer per IPv4 address.

---

14. Appendix B: Examples

TOC

14.1. Gateway based architecture

TOC

This architecture is targeted at residential broadband deployments but can be adapted easily to other types of deployment where the installed base of IPv4-only devices is important.

Consider a scenario where a Dual-Stack lite home gateway is provisioned only with IPv6 in the WAN port, no IPv4. The home gateway acts as an IPv4 DCHP server for the LAN network (wireline and wireless) handing out RFC1918 addresses. In addition, the home gateway may support IPv6 Auto-Configuration and/or DHCPv6 server for the LAN network. When an IPv4-only device connects to the home gateway, the gateway will hand it out a RFC1918 address. When a dual-stack capable device connects to the home gateway, the gateway will hand out a RFC1918 address and a global IPv6 address to the device. Besides, the home gateway will create an IPv4-in-IPv6 softwire tunnel [RFC5571] (Storer, B., Pignataro, C., Dos Santos, M., Stevant, B., Toutain, L., and J. Tremblay, "Softwire Hub and Spoke Deployment Framework with Layer Two Tunneling Protocol Version 2 (L2TPv2)," June 2009.)to an AFTR that resides in the service provider network.

When the device accesses IPv6 service, it will send the IPv6 datagram to the home gateway natively. The home gateway will route the traffic upstream to the default gateway.

When the device accesses IPv4 service, it will source the IPv4 datagram with the RFC1918 address and send the IPv4 datagram to the home gateway. The home gateway will encapsulate the IPv4 datagram inside the IPv4-in-IPv6 softwire tunnel and forward the IPv6 datagram to the AFTR. This contrasts what the home gateways normally do today which will NAT the RFC1918 address to the public IPv4 address and route the datagram upstream. When the AFTR receives the IPv6 datagram, it will decapsulate the IPv6 header and perform an IPv4-to-IPv4 NAT on the source address.

As illustrated in Figure 1 (gateway-based architecture), this dual-stack lite deployment model consists of three components: the dual-stack lite home router with a B4 element, the AFTR and a softwire between the B4 element acting as softwire initiator (SI) [RFC5571] (Storer, B., Pignataro, C., Dos Santos, M., Stevant, B., Toutain, L., and J. Tremblay, "Softwire Hub and Spoke Deployment Framework with Layer Two Tunneling Protocol Version 2 (L2TPv2)," June 2009.) in the dual-stack lite home router and the softwire concentrator (SC) [RFC5571] (Storer, B., Pignataro, C., Dos Santos, M., Stevant, B., Toutain, L., and J. Tremblay, "Softwire Hub and Spoke Deployment Framework with Layer Two Tunneling Protocol Version 2 (L2TPv2)," June 2009.) in the AFTR. The AFTR performs IPv4-IPv4 NAT translations to multiplex multiple subscribers through a pool of global IPv4 address. Overlapping address spaces used by subscribers are disambiguated through the identification of tunnel endpoints.

```
     +-----------+
     |    Host   |
     +-----+-----+
           |10.0.0.1
           |
           |
           |10.0.0.2
 +---------|---------+
 |         |         |
 |    Home router    |
 |+--------+--------+|
 ||       B4        ||
 |+--------+--------+|
 +--------|||--------+
          |||2001:0:0:1::1
          |||
          |||<-IPv4-in-IPv6 softwire
          |||
   -------|||-------
 /        |||        \
|   ISP core network  |
 \        |||        /
   -------|||-------
          |||
          |||2001:0:0:2::1
 +--------|||--------+
 |        AFTR       |
 |+--------+--------+|
 ||   Concentrator  ||
 |+--------+--------+|
 |       |NAT|       |
 |       +-+-+       |
 +---------|---------+
           |129.0.0.1
           |
   --------|--------
 /         |         \
|       Internet      |
 \         |         /
   --------|--------
           |
           |128.0.0.1
     +-----+-----+
     | IPv4 Host |
     +-----------+
```

Figure 1: gateway-based architecture

Notes:

*The dual-stack lite home router is not required to be on the same link as the host

*The dual-stack lite home router could be replaced by a dual-stack lite router in the service provider network

The resulting solution accepts an IPv4 datagram that is translated into an IPv4-in-IPv6 softwire datagram for transmission across the softwire. At the corresponding endpoint, the IPv4 datagram is decapsulated, and the translated IPv4 address is inserted based on a translation from the softwire.

14.1.1. Example message flow

TOC

In the example shown in Figure 2 (Outbound Datagram), the translation tables in the AFTR is configured to forward between IP/TCP (10.0.0.1/10000) and IP/TCP (129.0.0.1/5000). That is, a datagram received by the dual-stack lite home router from the host at address 10.0.0.1, using TCP DST port 10000 will be translated a datagram with IP SRC address 129.0.0.1 and TCP SRC port 5000 in the Internet.

```
             +-----------+
             |    Host   |
             +-----+-----+
                |  |10.0.0.1
IPv4 datagram 1 |  |
                |  |
                v  |10.0.0.2
         +---------|---------+
         |         |         |
         |    home router    |
         |+--------+--------+|
         ||        B4       ||
         |+--------+--------+|
         +--------|||--------+
                | |||2001:0:0:1::1
 IPv6 datagram 2| |||
                | |||<-IPv4-in-IPv6 softwire
                | |||
           -----|-|||-------
         /      | |||        \
        |   ISP core network  |
         \      | |||        /
           -----|-|||-------
                | |||
                | |||2001:0:0:2::1
         +------|-|||--------+
         |      | AFTR       |
         |      v |||        |
         |+--------+--------+|
         ||  Concentrartor  ||
         |+--------+--------+|
         |       |NAT|       |
         |       +-+-+       |
         +---------|---------+
                |  |129.0.0.1
IPv4 datagram 3 |  |
                |  |
           -----|--|--------
         /      |  |         \
        |       Internet      |
         \      |  |         /
           -----|--|--------
                |  |
                v  |128.0.0.1
             +-----+-----+
             | IPv4 Host |
             +-----------+
```

Figure 2: Outbound Datagram

Datagram	Header field	Contents
IPv4 datagram 1	IPv4 Dst	128.0.0.1
	IPv4 Src	10.0.0.1
	TCP Dst	80
	TCP Src	10000
---	---	---
IPv6 Datagram 2	IPv6 Dst	2001:0:0:2::1
	IPv6 Src	2001:0:0:1::1
	IPv4 Dst	128.0.0.1
	IPv4 Src	10.0.0.1
	TCP Dst	80
	TCP Src	10000
---	---	---
IPv4 datagram 3	IPv4 Dst	128.0.0.1
	IPv4 Src	129.0.0.1
	TCP Dst	80
	TCP Src	5000

Datagram header contents

When datagram 1 is received by the dual-stack lite home router, the B4 function encapsulates the datagram in datagram 2 and forwards it to the dual-stack lite carrier-grade NAT over the softwire.
When it receives datagram 2, the tunnel concentrator in the AFTR hands the IPv4 datagram to the NAT, which determines from its translation table that the datagram received on Softwire_1 with TCP SRC port 10000 should be translated to datagram 3 with IP SRC address 129.0.0.1 and TCP SRC port 5000.
Figure 3 (Inbound Datagram) shows an inbound message received at the AFTR. When the NAT function in the AFTR receives datagram 1, it looks up the IP/TCP DST in its translation table. In the example in Figure 3, the NAT translates the TCP DST port to 10000, sets the IP DST address to 10.0.0.1 and hands the datagram to the SC for transmission over Softwire_1. The B4 in the home router decapsulates IPv4 datagram from the inbound softwire datagram, and forwards it to the host.

```
             +-----------+
             |    Host   |
             +-----+-----+
                ^  |10.0.0.1
IPv4 datagram 3 |  |
                |  |
                |  |10.0.0.2
         +---------|---------+
         |       +-+-+       |
         |    home router    |
         |+--------+--------+|
         ||        B4       ||
         |+--------+--------+|
         +--------|||--------+
                ^ |||2001:0:0:1::1
IPv6 datagram 2 | |||
                | |||<-IPv4-in-IPv6 softwire
                | |||
           -----|-|||-------
         /      | |||        \
        |   ISP core network  |
         \      | |||        /
           -----|-|||-------
                | |||
                | |||2001:0:0:2::1
         +------|-|||--------+
         |       AFTR        |
         |+--------+--------+|
         ||   Concentrator  ||
         |+--------+--------+|
         |       |NAT|       |
         |       +-+-+       |
         +---------|---------+
                ^  |129.0.0.1
IPv4 datagram 1 |  |
                |  |
           -----|--|--------
         /      |  |         \
        |       Internet      |
         \      |  |         /
           -----|--|--------
                |  |
                |  |128.0.0.1
             +-----+-----+
             | IPv4 Host |
             +-----------+
```

Figure 3: Inbound Datagram

Datagram	Header field	Contents
IPv4 datagram 1	IPv4 Dst	129.0.0.1
	IPv4 Src	128.0.0.1
	TCP Dst	5000
	TCP Src	80
---	---	---
IPv6 Datagram 2	IPv6 Dst	2001:0:0:1::1
	IPv6 Src	2001:0:0:2::1
	IPv4 Dst	10.0.0.1
	IP Src	128.0.0.1
	TCP Dst	10000
	TCP Src	80
---	---	---
IPv4 datagram 3	IPv4 Dst	10.0.0.1
	IPv4 Src	128.0.0.1
	TCP Dst	10000
	TCP Src	80

Datagram header contents

14.1.2. Translation details

TOC

The AFTR has a NAT that translates between softwire/port pairs and IPv4-address/port pairs. The same translation is applied to IPv4 datagrams received on the device's external interface and from the softwire endpoint in the device.
In Figure 2 (Outbound Datagram), the translator network interface in the AFTR is on the Internet, and the softwire interface connects to the dual-stack lite home router. The AFTR translator is configured as follows:

> **Network interface:**

Translate IPv4 destination address and TCP destination port to the softwire identifier and TCP destination port

Softwire interface: Translate softwire identifier and TCP source port to IPv4 source address and TCP source port

Here is how the translation in Figure 3 (Inbound Datagram) works:

*Datagram 1 is received on the AFTR translator network interface. The translator looks up the IPv4-address/port pair in its translator table, rewrites the IPv4 destination address to 10.0.0.1 and the TCP source port to 10000, and hands the datagram to the SE to be forwarded over the softwire.

*The IPv4 datagram is received on the dual-stack lite home router B4. The B4 function extracts the IPv4 datagram and the dual-stack lite home router forwards datagram 3 to the host.

Softwire-Id/IPv4/Prot/Port	IPv4/Prot/Port
2001:0:0:1::1/10.0.0.1/TCP/10000	129.0.0.1/TCP/5000

Dual-Stack lite carrier-grade NAT translation table

The Softwire-Id is the IPv6 address assigned to the Dual-Stack lite home gateway. Hosts behind the same Dual-Stack lite home router have the same Softwire-Id. The source IPv4 is the RFC1918 addressed assigned by the Dual-Stack home router which is unique to each host behind the home gateway. The AFTR would receive packets sourced from different IPv4 addresses in the same softwire tunnel. The AFTR combines the Softwire-Id and IPv4 address/Port [Softwire-Id, IPv4+Port] to uniquely identify the host behind the same Dual-Stack lite home router.

14.2. Host based architecture

TOC

This architecture is targeted at new, large scale deployments of dual-stack capable devices implementing a dual-stack lite interface. Consider a scenario where a Dual-Stack lite host device is directly connected to the service provider network. The host device is dual-stack capable but only provisioned an IPv6 global address. Besides, the host device will pre-configure a well-known IPv4 non-routable address (see IANA section). This well-known IPv4 non-routable address is similar to the 127.0.0.1 loopback address. Every host device

implemented Dual-Stack lite will pre-configure the same address. This address will be used to source the IPv4 datagram when the device accesses IPv4 services. Besides, the host device will create an IPv4-in-IPv6 softwire tunnel to an AFTR. The Carrier Grade NAT will reside in the service provider network.

When the device accesses IPv6 service, the device will send the IPv6 datagram natively to the default gateway.

When the device accesses IPv4 service, it will source the IPv4 datagram with the well-known non-routable IPv4 address. Then, the host device will encapsulate the IPv4 datagram inside the IPv4-in-IPv6 softwire tunnel and send the IPv6 datagram to the AFTR. When the AFTR receives the IPv6 datagram, it will decapsulate the IPv6 header and perform IPv4-to-IPv4 NAT on the source address.

This scenario works on both wireline and wireless networks. A typical wireless device will connect directly to the service provider without home gateway in between.

As illustrated in Figure 4 (host-based architecture), this dual-stack lite deployment model consists of three components: the dual-stack lite host, the AFTR and a softwire between the softwire initiator B4 in the host and the softwire concentrator in the AFTR. The dual-stack lite host is assumed to have IPv6 service and can exchange IPv6 traffic with the AFTR.

The AFTR performs IPv4-IPv4 NAT translations to multiplex multiple subscribers through a pool of global IPv4 address. Overlapping IPv4 address spaces used by the dual-stack lite hosts are disambiguated through the identification of tunnel endpoints.

In this situation, the dual-stack lite host configures the IPv4 address 192.0.0.2 out of the well-known range 192.0.0.0/29 (defined by IANA) on its B4 interface. It also configure the first non-reserved IPv4 address of the reserved range, 192.0.0.1 as the address of its default gateway.

```
 +-------------------+
 |                   |
 |  Host 192.0.0.2   |
 |+--------+--------+|
 ||        B4       ||
 |+--------+--------+|
 +--------|||--------+
          |||2001:0:0:1::1
          |||
          |||<-IPv4-in-IPv6 softwire
          |||
   -------|||-------
 /        |||        \
|   ISP core network  |
 \        |||        /
   -------|||-------
          |||
          |||2001:0:0:2::1
 +--------|||--------+
 |       AFTR        |
 |+--------+--------+|
 ||  Concentrator   ||
 |+--------+--------+|
 |       |NAT|       |
 |       +-+-+       |
 +---------|---------+
           |129.0.0.1
           |
   --------|--------
 /         |         \
|       Internet      |
 \         |         /
   --------|--------
           |
           |128.0.0.1
     +-----+-----+
     | IPv4 Host |
     +-----------+
```

Figure 4: host-based architecture

The resulting solution accepts an IPv4 datagram that is translated into an IPv4-in-IPv6 softwire datagram for transmission across the softwire. At the corresponding endpoint, the IPv4 datagram is decapsulated, and

the translated IPv4 address is inserted based on a translation from the softwire.

14.2.1. Example message flow

TOC

In the example shown in Figure 5 (Outbound Datagram), the translation tables in the AFTR is configured to forward between IP/TCP (a.b.c.d/10000) and IP/TCP (129.0.0.1/5000). That is, a datagram received from the host at address 192.0.0.2, using TCP DST port 10000 will be translated a datagram with IP SRC address 129.0.0.1 and TCP SRC port 5000 in the Internet.

```
          +-------------------+
          |                   |
          |Host 192.0.0.2     |
          |+--------+--------+|
          ||        B4       ||
          |+--------+--------+|
          +--------|||--------+
                 | |||2001:0:0:1::1
 IPv6 datagram 1| |||
                 | |||<-IPv4-in-IPv6 softwire
                 | |||
            -----|-|||-------
          /      | |||        \
         |   ISP core network  |
          \      | |||        /
            -----|-|||-------
                 | |||
                 | |||2001:0:0:2::1
          +------|-|||--------+
          |      | AFTR       |
          |      v |||        |
          |+--------+--------+|
          ||  Concentrator   ||
          |+--------+--------+|
          |       |NAT|       |
          |       +-+-+       |
          +---------|---------+
                 |  |129.0.0.1
IPv4 datagram 2 |  |
            -----|--|--------
          /      |  |         \
         |       Internet      |
          \      |  |         /
            -----|--|--------
                 |  |
                 v  |128.0.0.1
              +-----+-----+
              | IPv4 Host |
              +-----------+
```

Figure 5: Outbound Datagram

Datagram	Header field	Contents
IPv6 Datagram 1	IPv6 Dst	2001:0:0:2::1
	IPv6 Src	2001:0:0:1::1
	IPv4 Dst	128.0.0.1
	IPv4 Src	a.b.c.d
	TCP Dst	80
	TCP Src	10000
---	---	---
IPv4 datagram 2	IPv4 Dst	128.0.0.1
	IPv4 Src	129.0.0.1
	TCP Dst	80
	TCP Src	5000

Datagram header contents

When sending an IPv4 packet, the dual-stack lite host encapsulates it in datagram 1 and forwards it to the AFTR over the softwire.
When it receives datagram 1, the concentrator in the AFTR hands the IPv4 datagram to the NAT, which determines from its translation table that the datagram received on Softwire_1 with TCP SRC port 10000 should be translated to datagram 3 with IP SRC address 129.0.0.1 and TCP SRC port 5000.
Figure 6 (Inbound Datagram) shows an inbound message received at the AFTR. When the NAT function in the AFTR receives datagram 1, it looks up the IP/TCP DST in its translation table. In the example in Figure 3, the NAT translates the TCP DST port to 10000, sets the IP DST address to a.b.c.d and hands the datagram to the concentrator for transmission over Softwire_1. The B4 in the dual-stack lite hosts decapsulates IPv4 datagram from the inbound softwire datagram, and forwards it to the host.

```
          +-------------------+
          |                   |
          |Host 192.0.0.2     |
          |+--------+--------+|
          ||        B4       ||
          |+--------+--------+|
          +--------|||--------+
                 ^ |||2001:0:0:1::1
IPv6 datagram 2 | |||
                | |||<-IPv4-in-IPv6 softwire
                | |||
            -----|-|||-------
          /      | |||        \
         |   ISP core network  |
          \      | |||        /
            -----|-|||-------
                 | |||
                 | |||2001:0:0:2::1
          +------|-|||--------+
          |       AFTR        |
          |      | |||        |
          |+--------+--------+|
          ||  Concentrator   ||
          |+--------+--------+|
          |       |NAT|       |
          |       +-+-+       |
          +---------|---------+
                 ^  |129.0.0.1
IPv4 datagram 1 |  |
            -----|--|--------
          /      |  |         \
         |       Internet      |
          \      |  |         /
            -----|--|--------
                 |  |
                 |  |128.0.0.1
              +-----+-----+
              | IPv4 Host |
              +-----------+
```

Figure 6: Inbound Datagram

Datagram	Header field	Contents
IPv4 datagram 1	IPv4 Dst	129.0.0.1
	IPv4 Src	128.0.0.1
	TCP Dst	5000
	TCP Src	80
---	---	---
IPv6 Datagram 2	IPv6 Dst	2001:0:0:1::1
	IPv6 Src	2001:0:0:2::1
	IPv4 Dst	a.b.c.d
	IP Src	128.0.0.1
	TCP Dst	10000
	TCP Src	80

Datagram header contents

14.2.2. Translation details

TOC

The translations happening in the AFTR are the same as in the previous examples. The well known IPv4 address 192.0.0.2 out of the 192.0.0.0/29 (defined by IANA) range used by all the hosts are disambiguated by the IPv6 source address of the softwire.

Softwire-Id/IPv4/Prot/Port	IPv4/Prot/Port
2001:0:0:1::1/a.b.c.d/TCP/10000	129.0.0.1/TCP/5000

Dual-Stack lite carrier-grade NAT translation table

The Softwire-Id is the IPv6 address assigned to the Dual-Stack host. Each host has an unique Softwire-Id. The source IPv4 address is one of the well-known IPv4 address. The AFTR could receive packets from different hosts sourced from the same IPv4 well-known address from different softwire tunnels. Similar to the gateway architecture, the AFTR combines the Softwire-Id and IPv4 address/Port [Softwire-Id, IPv4+Port] to uniquely identify the individual host.

---

15. Appendix C: Deployment considerations

TOC

15.1. AFTR service distribution and horizontal scaling

TOC

One of the key benefits of the dual-stack lite technology lies in the fact it is tunnel based. That is, tunnel end-points may be anywhere in the service provider network.
Using the DHCPv6 tunnel end-point option, service providers can create groups of users sharing the same AFTR. Those groups can be merged or divided at will. This leads to an horizontally scaled solution, where more capacity is added simply by adding more boxes. As those groups of users can evolve over time, it is best to make sure that AFTRs do not require per-user configuration in order to provide service.

15.2. Horizontal scaling

TOC

A service provider can start using just a few AFTR centrally located. Later, when more capacity is needed, more boxes can be added and pushed to the edges of the access network. In case of a spike of traffic, for example during the Olympic games or an important political event, capacity can be quickly added in any location of the network (tunnels can terminate anywhere) simply by splitting user groups. Extra capacity can be later removed when the traffic returns to normal by resetting the DHCPv6 tunnel end-point settings.

15.3. High availability

TOC

An important element in the design of the dual-stack lite technology is the simplicity of implementation on the customer side. A simple IP4-in-IPv6 tunnel and a default route over it is all is needed to get IPv4 connectivity. Dealing with high availability is the responsibility of the service provider, not the customer devices implementing dual-stack lite. As such, a single IPv6 address of the tunnel end-point is provided in the DHCPv6 option defined in [I-D.ietf-softwire-ds-lite-tunnel-option] (Hankins, D. and T. Mrugalski, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Options for Dual- Stack Lite,” March 2010.). The service provider can use techniques such as anycast or various types of clusters to ensure

availability of the IPv4 service. The exact synchronization (or lack thereof) between redundant AFTRs is out of scope for this document.

15.4. Logging

TOC

DS-Lite AFTR implementation should offer the possility to log NAT binding creations or other ways to keep track of the ports/IP addresses used by customers. This is both to support troubleshooting, which is very important to service providers trying to figure out why something may not be working, as well as to meet region-specific requirements for responding to legally-binding requests for information from law enforcement authorities.

16. References

TOC

16.1. Normative references

TOC

[I-D.ietf-softwire-ds-lite-tunnel-option]	Hankins, D. and T. Mrugalski, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Options for Dual- Stack Lite,” draft-ietf-softwire-ds-lite-tunnel-option-02 (work in progress), March 2010 (TXT).
[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2473]	Conta, A. and S. Deering, “Generic Packet Tunneling in IPv6 Specification,” RFC 2473, December 1998 (TXT, HTML, XML).
[RFC2474]	Nichols, K., Blake, S., Baker, F., and D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,” RFC 2474, December 1998 (TXT, HTML, XML).
[RFC4213]	Nordmark, E. and R. Gilligan, “Basic Transition Mechanisms for IPv6 Hosts and Routers,” RFC 4213, October 2005 (TXT).
[RFC5625]	Bellis, R., “DNS Proxy Implementation Guidelines,” BCP 152, RFC 5625, August 2009 (TXT).

16.2. Informative references

TOC

[I-D.bajko-v6ops-port-restricted-ipaddr-assign]	Bajko, G. and T. Savolainen, "Port Restricted IP Address Assignment," draft-bajko-v6ops-port-restricted-ipaddr-assign-02 (work in progress), November 2008 (TXT).
[I-D.cheshire-nat-pmp]	Cheshire, S., "NAT Port Mapping Protocol (NAT-PMP)," draft-cheshire-nat-pmp-03 (work in progress), April 2008 (TXT).
[I-D.droms-softwires-snat]	Droms, R. and B. Haberman, "Softwires Network Address Translation (SNAT)," draft-droms-softwires-snat-01 (work in progress), July 2008 (TXT).
[I-D.durand-dual-stack-lite]	Durand, A., "Dual-stack lite broadband deployments post IPv4 exhaustion," draft-durand-dual-stack-lite-00 (work in progress), July 2008 (TXT).
[I-D.ford-shared-addressing-issues]	Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing," draft-ford-shared-addressing-issues-02 (work in progress), March 2010 (TXT).
[I-D.nishitani-cgn]	Yamagata, I., Nishitani, T., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for IP address sharing schemes," draft-nishitani-cgn-04 (work in progress), March 2010 (TXT).
[I-D.templin-seal]	Templin, F., "The Subnetwork Encapsulation and Adaptation Layer (SEAL)," draft-templin-seal-23 (work in progress), August 2008 (TXT).
[I-D.ymbk-aplusp]	Bush, R., "The A+P Approach to the IPv4 Address Shortage," draft-ymbk-aplusp-05 (work in progress), October 2009 (TXT).
[RFC1191]	Mogul, J. and S. Deering, "Path MTU discovery," RFC 1191, November 1990 (TXT).
[RFC1918]	Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets," BCP 5, RFC 1918, February 1996 (TXT).
[RFC2663]	Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," RFC 2663, August 1999 (TXT).
[RFC2993]	Hain, T., "Architectural Implications of NAT," RFC 2993, November 2000 (TXT).
[RFC4787]	Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP," BCP 127, RFC 4787, January 2007 (TXT).

[RFC5382]	Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP," BCP 142, RFC 5382, October 2008 (TXT).
[RFC5508]	Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT Behavioral Requirements for ICMP," BCP 148, RFC 5508, April 2009 (TXT).
[RFC5571]	Storer, B., Pignataro, C., Dos Santos, M., Stevant, B., Toutain, L., and J. Tremblay, "Softwire Hub and Spoke Deployment Framework with Layer Two Tunneling Protocol Version 2 (L2TPv2)," RFC 5571, June 2009 (TXT).
[UPnP-IGD]	UPnP Forum, "Universal Plug and Play Internet Gateway Device Standardized Gateway Device Protocol," September 2006.

Author's Address

TOC

	Alain Durand (editor)
	Comcast
	1, Comcast center
	Philadelphia, PA 19103
	USA
Email:	alain_durand@cable.comcast.com