Internet Engineering Task Force (IETF)
Request for Comments: 6333
Category: Standards Track
ISSN: 2070-1721

A. Durand
Juniper Networks
R. Droms
Cisco
J. Woodyatt
Apple
Y. Lee
Comcast

August 2011

Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion

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 in service provider networks. 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 is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6333.

Table of Contents

1. Introduction

The common thinking for more than 10 years has been that the transition to IPv6 will be based solely 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 has now been depleted, well before sufficient IPv6 deployment had taken place. As a result, many IPv4 services have to continue to be provided even under severely limited address space.

This document specifies the Dual-Stack Lite technology, which is aimed at better aligning the costs and benefits in service provider networks. Dual-Stack Lite will enable both continued support for IPv4 services and incentives for the deployment of IPv6. It also de-couples IPv6 deployment in the service provider network from the rest of the Internet, making incremental deployment easier.

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).

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.

This document will first present some deployment scenarios and then define the behavior of the two elements of the Dual-Stack Lite technology: the Basic Bridging BroadBand (B4) element and the Address Family Transition Router (AFTR) element. It will then go into networking and NAT-ing considerations.

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 RFC 2119 [RFC2119].

3. Terminology

The technology described in this document is known as Dual-Stack Lite. The abbreviation "DS-Lite" will be used throughout this text.

This document also introduces two new terms: the DS-Lite Basic Bridging BroadBand (B4) element and the DS-Lite Address Family Transition Router (AFTR) element.

Dual-stack is defined in [RFC4213].

NAT-related terminology is defined in [RFC4787].

CPE stands for Customer Premise Equipment. This is the layer 3 device in the customer premise that is connected to the service provider network. That device is often a home gateway. However, sometimes computers are directly attached to the service provider network. In such cases, such computers can be viewed as CPEs as well.

4. Deployment Scenarios

4.1. Access Model

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 a number 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 and 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 lends itself to horizontal scaling and provides some flexibility to adapt to changing traffic load. More discussion of horizontal scaling can be found in Appendix A.

- 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, and 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 payloads.

4.2. CPE

This section describes home Local Area networks characterized by the presence of a home gateway, or CPE, provisioned only with IPv6 by the service provider.

A DS-Lite CPE is an IPv6-aware CPE with a B4 interface implemented in the WAN interface.

A DS-Lite CPE SHOULD NOT operate a NAT function between an internal interface and a B4 interface, as the NAT function will be performed by the AFTR in the service provider's network. This will avoid accidentally operating in a double-NAT environment.

However, it SHOULD operate its own DHCP(v4) server handing out [RFC1918] 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 CPE will forward those DNS requests via the B4 interface, the same way it forwards any regular IPv4 packets. However, each DNS request will create a binding in the AFTR. A large number of DNS requests may have a direct impact on the AFTR's NAT table utilization.

IPv6-capable devices directly reach the IPv6 Internet. Packets simply follow IPv6 routing, they do not go through the tunnel, and they 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

In broadband home networks, some devices are directly connected to the broadband service provider. They are connected straight to a modem, without a home gateway. Those devices are, in fact, acting as CPEs.

Under this scenario, the customer device is a dual-stack-capable host that is provisioned by the service provider with IPv6 only. 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/CPE 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 they 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

5.1. Definition

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

5.2. Encapsulation

The tunnel is a multipoint-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

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] 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 the extra IPv6 header. The original IPv4 packet is not oversized. The packet is oversized after the IPv6 encapsulation. The inner IPv4 packet MUST NOT be fragmented. Fragmentation MUST happen after the encapsulation of the IPv6 packet. Reassembly MUST happen before the decapsulation of the IPv4 packet. A detailed procedure has been specified in [RFC2473] Section 7.2.

5.4. AFTR Discovery

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 [RFC6334].

5.5. DNS

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].

To support a security-aware resolver behind the B4 element, the DNS proxy in the B4 element must also be security aware. Details can be found in [RFC4033] Section 6.

5.6. Interface Initialization

The B4 element can be implemented in a host and CPE in conjunction with other technologies such as native dual-stack. The host and the CPE SHOULD select to start only one technology during initialization. For example, if the CPE selects to start in native dual-stack mode, it SHOULD NOT initialize the B4 element. This selection process is out of scope for this document.

5.7. Well-Known IPv4 Address

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, and 192.0.0.2 is reserved for the B4 element. If a service provider has a special configuration that prevents the B4 element from using 192.0.0.2, 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

6.1. Definition

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

6.2. Encapsulation

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

As noted previously, fragmentation and reassembly need to be taken care of by the tunnel endpoints. As such, the AFTR MUST perform fragmentation and reassembly if the underlying link MTU cannot accommodate the encapsulation overhead. Fragmentation MUST happen after the encapsulation on the IPv6 packet. Reassembly MUST happen before the decapsulation of the IPv6 header. A detailed procedure has been specified in [RFC2473] Section 7.2.

Fragmentation at the Tunnel Entry-Point is a lightweight 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 devices simultaneously source a large number of fragmented packets through the AFTR to its managed B4 elements, 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’s performance. However, this impact only happens when many clients simultaneously source large IPv4 packets. Since we believe that the 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), reassembly is only a fraction of the overall AFTR’s workload.

When the AFTR's resources are running below a pre-defined threshold, the AFTR SHOULD generate a notification to the administrator before the resources are completely exhausted. The threshold and notification procedures are implementation dependent and are out of scope for this document.

Methods to avoid fragmentation, such as rewriting the TCP Maximum Segment Size (MSS) option or using technologies such as the Subnetwork Encapsulation and Adaptation Layer as defined in [RFC5320], are out of scope for this document.

6.4. DNS

As noted previously, a DS-Lite node implementing a B4 element will perform DNS resolution over IPv6. As a result, DNS packets are not expected to go through the AFTR element.

6.5. Well-Known IPv4 Address

The AFTR SHOULD 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

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 lookup in the extended IPv4 NAT binding table, the AFTR knows how to reconstruct the IPv6 encapsulation when the packets come back from the Internet. That way, there is no need to keep a static configuration for each tunnel.

7. Network Considerations

7.1. Tunneling

Tunneling MUST be done in accordance to [RFC2473] and [RFC4213]. Traffic classes ([RFC2474]) from the IPv4 headers MUST be carried over to the IPv6 headers and vice versa.

7.2. Multicast Considerations

Discussion of multicast is out of scope for this document.

8. NAT Considerations

8.1. NAT Pool

The AFTR MAY be provisioned with different NAT pools. The address ranges in the pools may be disjoint but MUST NOT be overlapped. Operators may implement policies in the AFTR to assign clients in different pools. For example, an AFTR can have two interfaces. Each interface will have a disjoint pool NAT assigned to it. In another case, a policy implemented on the AFTR may specify that one set of B4s will use NAT pool 1 and a different set of B4s will use NAT pool 2.

8.2. NAT Conformance

A Dual-Stack Lite AFTR MUST implement behavior conforming to the best current practice, currently documented in [RFC4787], [RFC5508], and [RFC5382]. More discussions about carrier-grade NATs can be found in [LSN-REQS].

8.3. Application Level Gateways (ALGs)

The AFTR performs NAT-44 and inherits the limitations of NAT. Some protocols require ALGs in the NAT device to traverse through the NAT. For example, Active FTP requires the ALG to work properly. ALGs consume resources, and there are many different types of ALGs. The AFTR is a shared network device that supports a large number of B4 elements. It is impossible for the AFTR to implement every current and future ALG.

8.4. Sharing Global IPv4 Addresses

The AFTR shares a single IP with multiple users. This helps to increase the IPv4 address utilization. However, it also brings some issues such as logging and lawful intercept. More considerations on sharing the port space of IPv4 addresses can be found in [RFC6269].

8.5. Port Forwarding / Keep Alive

The PCP working group is standardizing a control plane to the carrier-grade NAT [LSN-REQS] in the IETF. The Port Control Protocol (PCP) enables applications to directly negotiate with the NAT to open ports and negotiate lifetime values to avoid keep-alive traffic. More on PCP can be found in [PCP-BASE].

9. Acknowledgements

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 [SNAT]. Note that this document results from a merging of [DURAND-DS-LITE] and [SNAT]. 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, and Ichiro Mizukoshi. The authors 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

Per this document, IANA has allocated a well-known 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

Security issues associated with NAT have long been documented. See [RFC2663] and [RFC2993].

However, moving the NAT functionality from the CPE 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 timestamps are no longer sufficient to identify a particular broadband customer. The AFTR should have the capability to log the tunnel-id, protocol, ports/IP addresses, and the creation time of the NAT binding to uniquely identify the user sessions. Exact details of what is logged are implementation specific and out of scope for this document.

The AFTR performs translation functions for interior IPv4 hosts using RFC 1918 addresses or the IANA reserved address range (192.0.0.0/29). In some circumstances, an ISP may provision policies in the AFTR and instruct the AFTR to bypass translation functions based on <IPv4 Address, port number, protocol>. When the AFTR receives a packet with matching information of the policy from the interior host, the AFTR can simply forward the packet without translation. The addresses, ports, and protocol information must be provisioned on the AFTR before receiving the packet. The provisioning mechanism is out of scope for this specification.

When decapsulating packets, the AFTR MUST only forward packets sourced by RFC 1918 addresses, an IANA reserved address range, or any other out-of-band pre-authorized addresses. The AFTR MUST drop all other packets. This prevents rogue devices from launching denial-of-service attacks using unauthorized public IPv4 addresses in the IPv4 source header field or an unauthorized transport port range in the IPv4 transport header field. For example, rogue devices could bombard a public web server by launching a TCP SYN ACK attack [RFC4987]. 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 ports usable by a 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 sharing IPv4 addresses can be found in [RFC6269]. Other considerations and recommendations on logging can be found in [RFC6302].

AFTRs should support ways to limit service only to registered customers. One simple option is to implement an IPv6 ingress filter on the AFTR's tunnel interface to accept only the IPv6 address range defined in the filter.

12. References

12.1. Normative References

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

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

[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.

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

[RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, August 2009.

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

12.2. Informative References

[DURAND-DS-LITE] Durand, A., "Dual-stack lite broadband deployments post IPv4 exhaustion", Work in Progress, July 2008.

[LSN-REQS] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NAT (CGN)", Work in Progress, July 2011.

[PCP-BASE] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", Work in Progress, July 2011.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

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

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, November 2000.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4787] Audet, F., Ed., and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

[RFC4987] Eddy, W., "TCP SYN Flooding Attacks and Common Mitigations", RFC 4987, August 2007.

[RFC5320] Templin, F., Ed., "The Subnetwork Encapsulation and Adaptation Layer (SEAL)", RFC 5320, February 2010.

[RFC5382] Guha, S., Ed., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.

[RFC5508] Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT Behavioral Requirements for ICMP", BCP 148, RFC 5508, April 2009.

[RFC5571] Storer, B., Pignataro, C., Ed., Dos Santos, M., Stevant, B., Ed., Toutain, L., and J. Tremblay, "Softwire Hub and Spoke Deployment Framework with Layer Two Tunneling Protocol Version 2 (L2TPv2)", RFC 5571, June 2009.

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

[RFC6302] Durand, A., Gashinsky, I., Lee, D., and S. Sheppard, "Logging Recommendations for Internet-Facing Servers", BCP 162, RFC 6302, June 2011.

[SNAT] Droms, R. and B. Haberman, "Softwires Network Address Translation (SNAT)", Work in Progress, July 2008.

Appendix A. Deployment Considerations

A.1. AFTR Service Distribution and Horizontal Scaling

One of the key benefits of the Dual-Stack Lite technology lies in the fact that it is a tunnel-based solution. As such, tunnel endpoints can be anywhere in the service provider network.

Using the DHCPv6 tunnel endpoint option [RFC6334], service providers can create groups of users sharing the same AFTR. Those groups can be merged or divided at will. This leads to a horizontally scaled solution, where more capacity is added with more AFTRs. 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.

A.2. Horizontal Scaling

A service provider can start using just a few centralized AFTRs. Later, when more capacity is needed, more AFTRs can be added and pushed closer to the edges of the access network.

A.3. High Availability

An important element in the design of the Dual-Stack Lite technology is the simplicity of implementation on the customer side. An IP4-in-IPv6 tunnel and a default route over it in the B4 element are all that is needed to get IPv4 connectivity. It is assumed that 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 endpoint is provided in the DHCPv6 option defined in [RFC6334]. Specific means to achieve high availability on the service provider side are outside the scope of this specification.

A.4. Logging

DS-Lite AFTR implementation should offer the functionality 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, and to meet region-specific requirements for responding to legally binding requests for information from law enforcement authorities.

Appendix B. Examples

B.1. Gateway-Based Architecture

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 CPE is provisioned only with IPv6 in the WAN port, not IPv4. The CPE acts as an IPv4 DHCP server for the LAN (wireline and wireless) handing out [RFC1918] addresses. In addition, the CPE may support IPv6 Auto-Configuration and/or a DHCPv6 server for the LAN. When an IPv4-only device connects to the CPE, that CPE will hand out a [RFC1918] address to the device. When a dual-stack-capable device connects to the CPE, that CPE will hand out a [RFC1918] address and a global IPv6 address to the device. Besides, the CPE will create an IPv4-in-IPv6 softwire tunnel [RFC5571] to an AFTR that resides in the service provider network.

When the device accesses IPv6 service, it will send the IPv6 datagram to the CPE natively. The CPE will route the traffic upstream to the IPv6 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 CPE. The CPE will encapsulate the IPv4 datagram inside the IPv4-in-IPv6 softwire tunnel and forward the IPv6 datagram to the AFTR. This is in contrast to what the CPE normally does today, which is to 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, 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] in the Dual-Stack Lite home router and the softwire concentrator (SC) [RFC5571] in the AFTR. The AFTR performs IPv4-IPv4 NAT translations to multiplex multiple subscribers through a pool of global IPv4 addresses. Overlapping address spaces used by subscribers are disambiguated through the identification of tunnel endpoints.

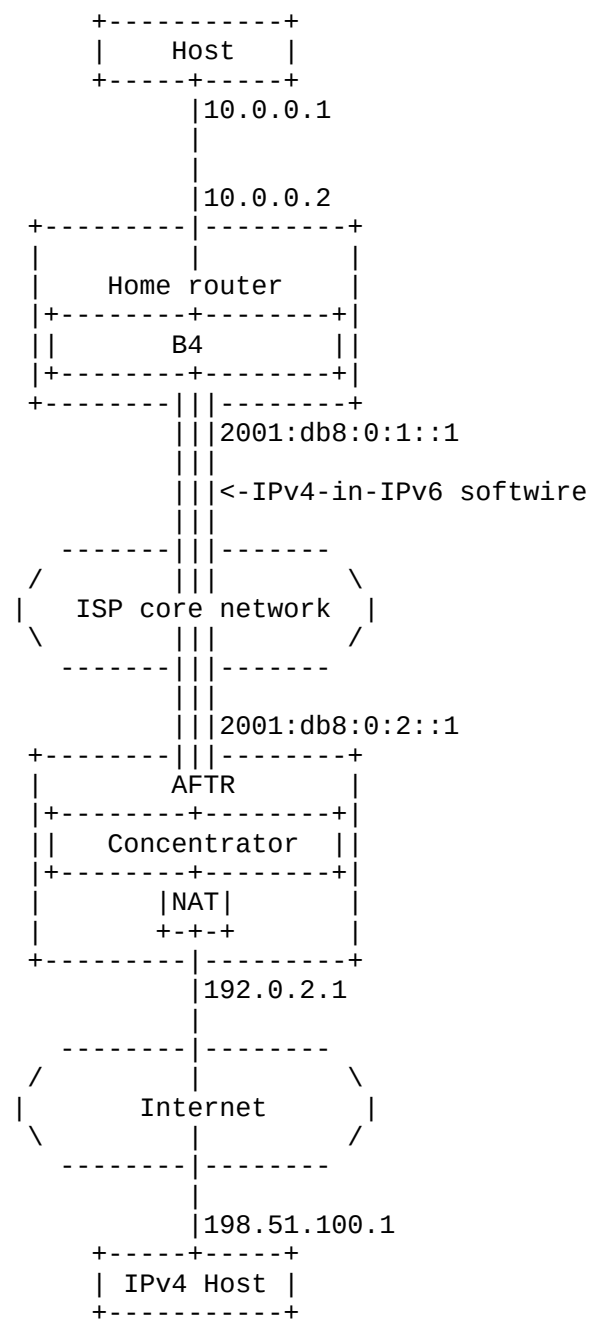

Figure 1: Gateway-Based Architecture

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.

B.1.1. Example Message Flow

In the example shown in Figure 2, the translation tables in the AFTR are configured to forward between IP/TCP (10.0.0.1/10000) and IP/TCP (192.0.2.1/5000). 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 to a datagram with IPv4 SRC address 192.0.2.1 and TCP SRC port 5000 in the Internet.

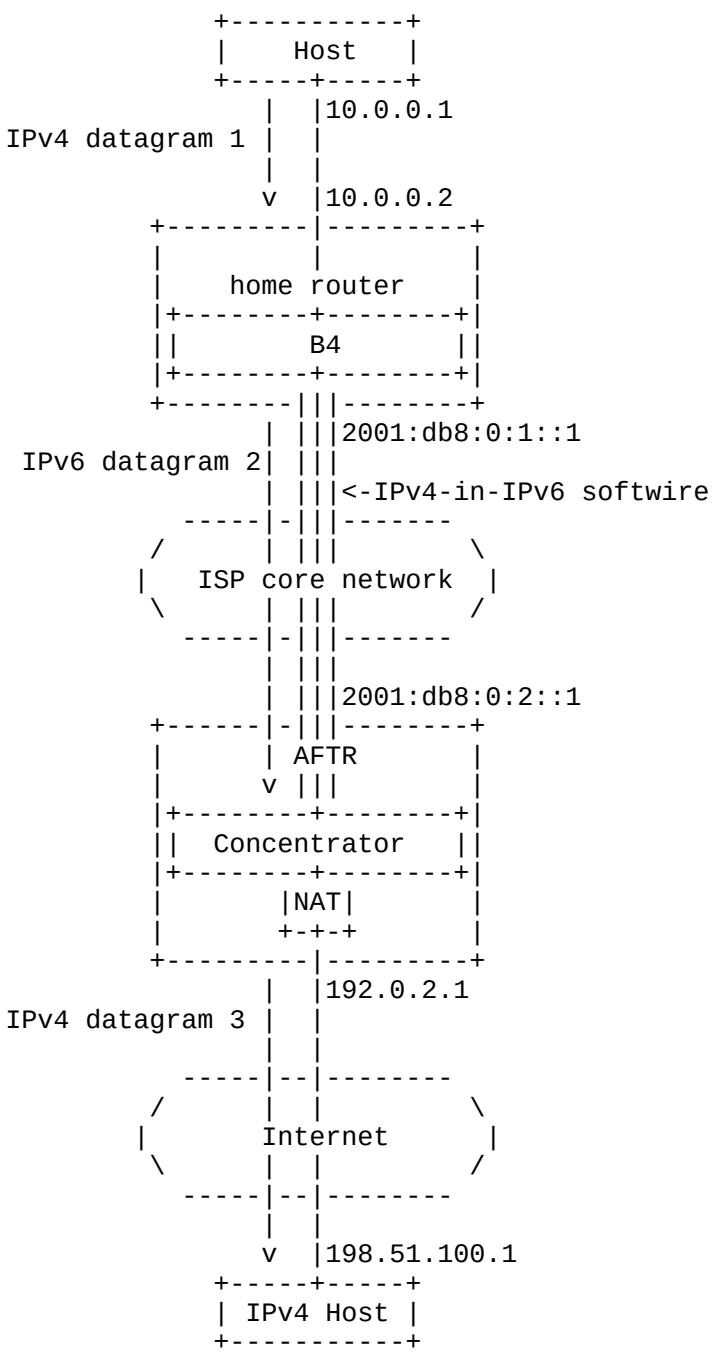

Figure 2: Outbound Datagram

Datagram	Header field	Contents
IPv4 datagram 1	IPv4 Dst	198.51.100.1
	IPv4 Src	10.0.0.1
	TCP Dst	80
	TCP Src	10000
---	---	---
IPv6 datagram 2	IPv6 Dst	2001:db8:0:2::1
	IPv6 Src	2001:db8:0:1::1
	IPv4 Dst	198.51.100.1
	IPv4 Src	10.0.0.1
	TCP Dst	80
	TCP Src	10000
---	---	---
IPv4 datagram 3	IPv4 Dst	198.51.100.1
	IPv4 Src	192.0.2.1
	TCP Dst	80
	TCP Src	5000

Datagram Header Contents

When datagram 1 is received by the Dual-Stack Lite home router, the B4 element encapsulates the datagram in datagram 2 and forwards it to the Dual-Stack Lite carrier-grade NAT over the softwire.

When the tunnel concentrator in the AFTR receives datagram 2, it forwards the IPv4 datagram to the NAT, which determines from its NAT table that the datagram received on the softwire with TCP SRC port 10000 should be translated to datagram 3 with IPv4 SRC address 192.0.2.1 and TCP SRC port 5000.

Figure 3 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 information in its translation table. In the example in Figure 3, the NAT changes the TCP DST port to 10000, sets the IP DST address to 10.0.0.1, and forwards the datagram to the softwire. The B4 in the home router decapsulates the IPv4 datagram from the inbound softwire datagram and forwards it to the host.

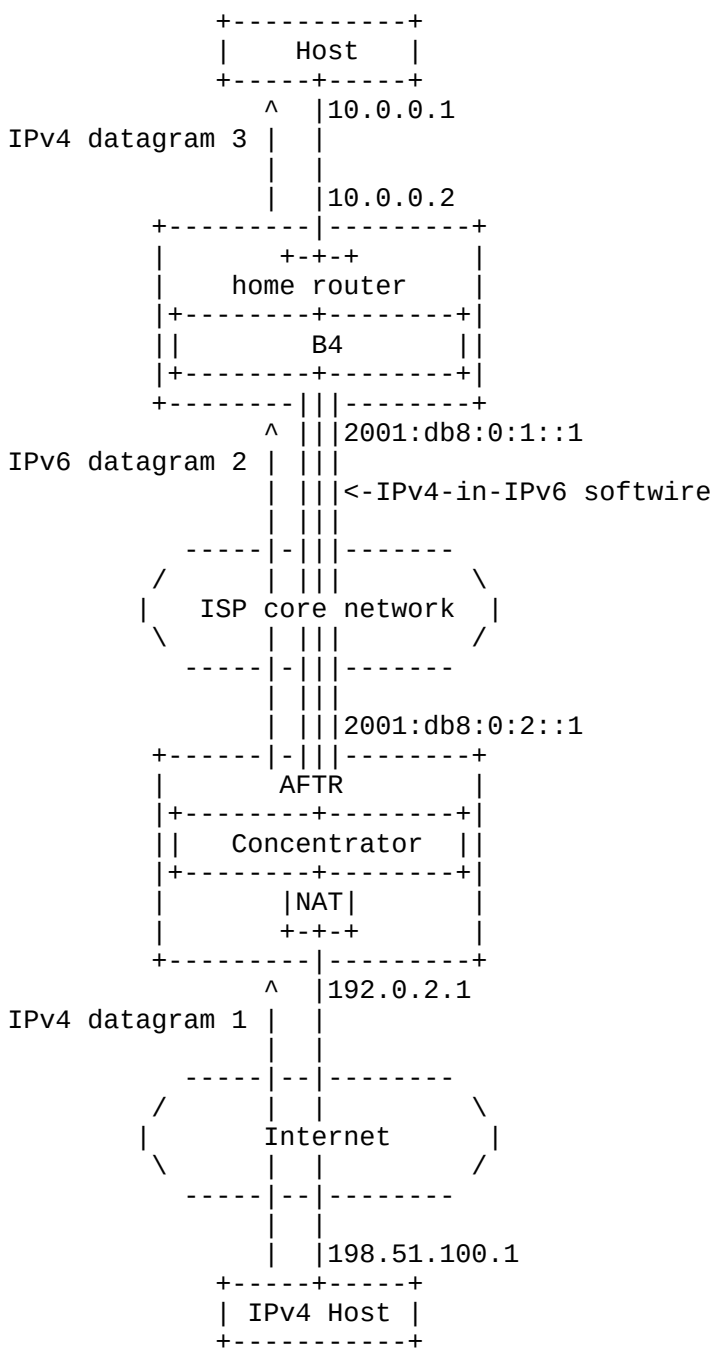

Figure 3: Inbound Datagram

Datagram	Header field	Contents
IPv4 datagram 1	IPv4 Dst	192.0.2.1
	IPv4 Src	198.51.100.1
	TCP Dst	5000
	TCP Src	80
---	---	---
IPv6 datagram 2	IPv6 Dst	2001:db8:0:1::1
	IPv6 Src	2001:db8:0:2::1
	IPv4 Dst	10.0.0.1
	IPv4 Src	198.51.100.1
	TCP Dst	10000
	TCP Src	80
---	---	---
IPv4 datagram 3	IPv4 Dst	10.0.0.1
	IPv4 Src	198.51.100.1
	TCP Dst	10000
	TCP Src	80

Datagram Header Contents

B.1.2. Translation Details

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, 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 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 forwards the datagram to the softwire.

o 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:db8:0:1::1/10.0.0.1/TCP/10000	192.0.2.1/TCP/5000

Dual-Stack Lite Carrier-Grade NAT Translation Table

The Softwire-Id is the IPv6 address assigned to the Dual-Stack Lite CPE. Hosts behind the same Dual-Stack Lite home router have the same Softwire-Id. The source IPv4 address is the [RFC1918] address assigned by the Dual-Stack home router and is unique to each host behind the CPE. 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.

B.2. Host-Based Architecture

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 with an IPv6 global address. Besides, the host device will pre-configure a well-known IPv4 non-routable address; see Section 10 (IANA Considerations). This well-known IPv4 non-routable address is similar to the 127.0.0.1 loopback address. Every host device that implements 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 a CPE in between.

As illustrated in Figure 4, 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 addresses. 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 configures the first non-reserved IPv4 address of the reserved range, 192.0.0.1, as the address of its default gateway.

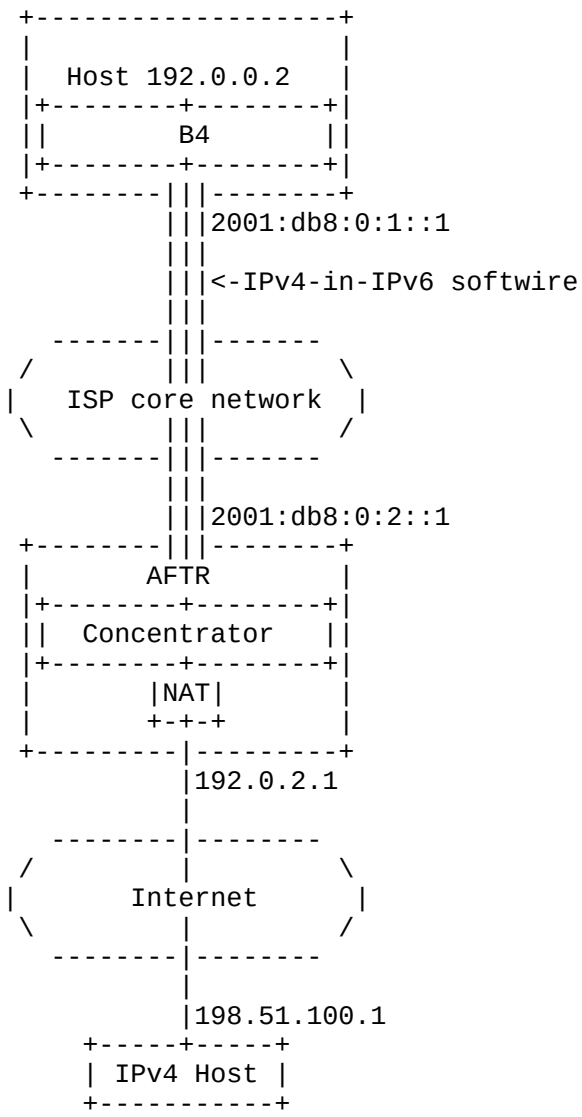

Figure 4: Host-Based Architecture

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.

B.2.1. Example Message Flow

In the example shown in Figure 5, the translation tables in the AFTR are configured to forward between IP/TCP (192.0.0.2/10000) and IP/TCP (192.0.2.1/5000). That is, a datagram received from the host at address 192.0.0.2, using TCP DST port 10000, will be translated to a datagram with IPv4 SRC address 192.0.2.1 and TCP SRC port 5000 in the Internet.

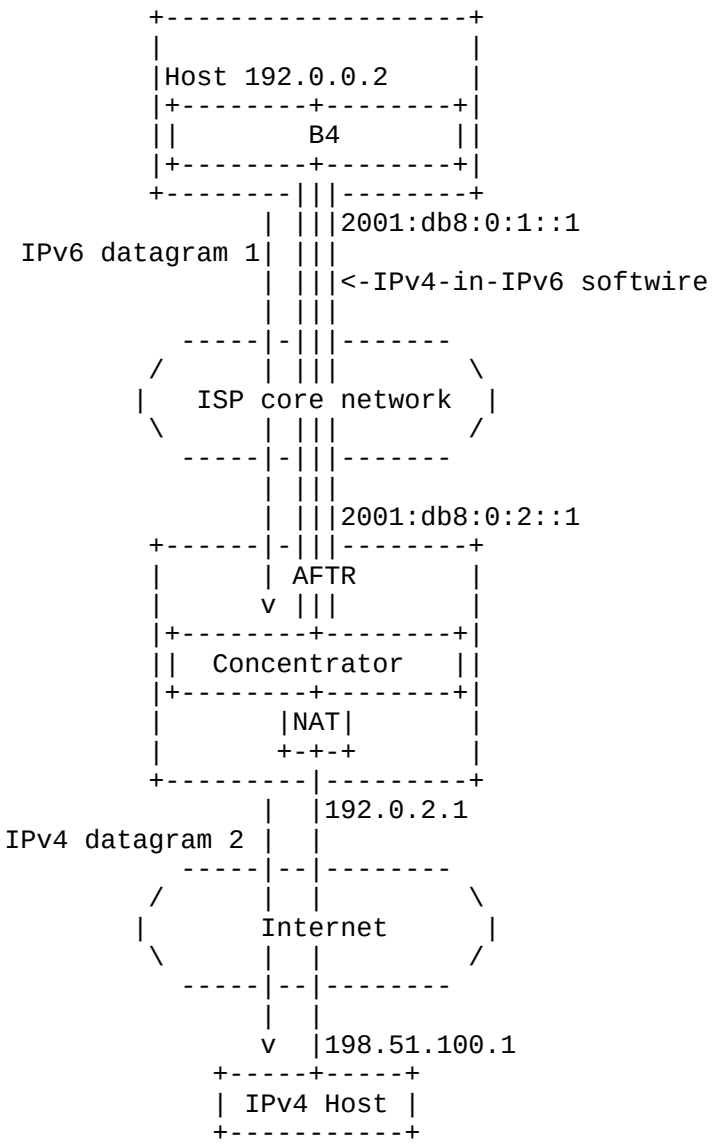

Figure 5: Outbound Datagram

Datagram	Header field	Contents
IPv6 datagram 1	IPv6 Dst	2001:db8:0:2::1
	IPv6 Src	2001:db8:0:1::1
	IPv4 Dst	198.51.100.1
	IPv4 Src	192.0.0.2
	TCP Dst	80
	TCP Src	10000
---	---	---
IPv4 datagram 2	IPv4 Dst	198.51.100.1
	IPv4 Src	192.0.2.1
	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 the softwire with TCP SRC port 10000 should be translated to datagram 3 with IPv4 SRC address 192.0.2.1 and TCP SRC port 5000.

Figure 6 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 6, the NAT translates the TCP DST port to 10000, sets the IP DST address to 192.0.0.2, and forwards the datagram to the softwire. The B4 inside the host decapsulates the IPv4 datagram from the inbound softwire datagram, and forwards it to the host’s application layer.

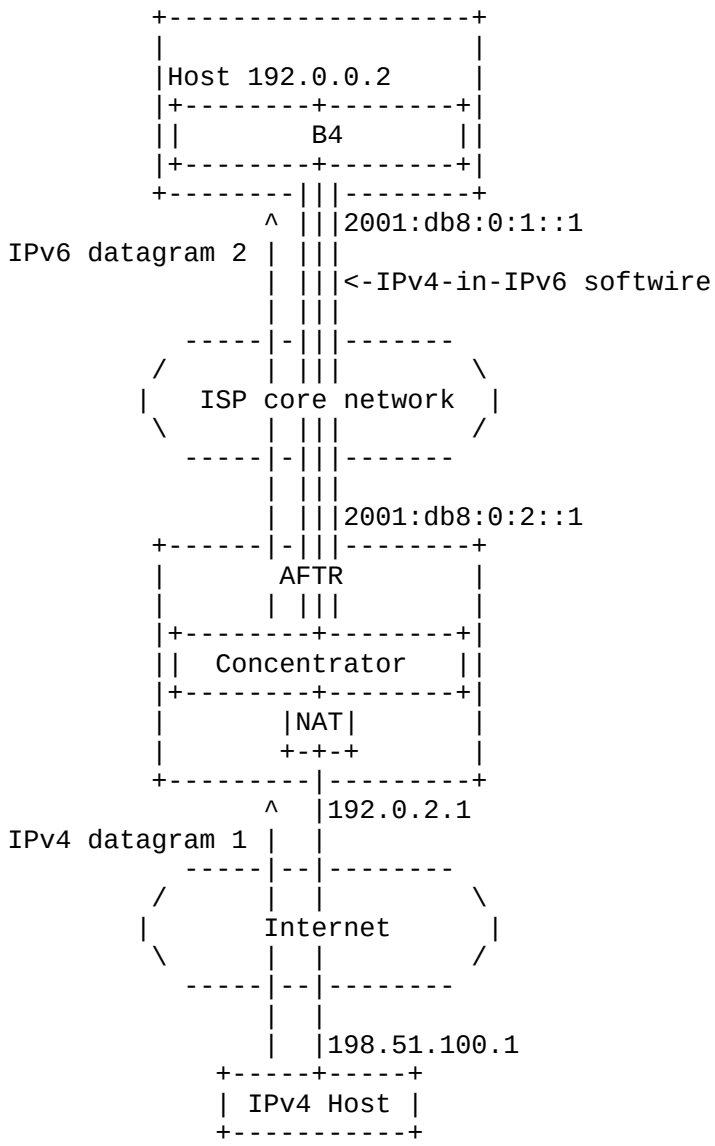

Figure 6: Inbound Datagram

Datagram	Header field	Contents
IPv4 datagram 1	IPv4 Dst	192.0.2.1
	IPv4 Src	198.51.100.1
	TCP Dst	5000
	TCP Src	80
---	---	---
IPv6 datagram 2	IPv6 Dst	2001:db8:0:1::1
	IPv6 Src	2001:db8:0:2::1
	IPv4 Dst	192.0.0.2
	IPv4 Src	198.51.100.1
	TCP Dst	10000
	TCP Src	80

Datagram Header Contents

B.2.2. Translation Details

The AFTR translation steps are the same as in Appendix B.1.2. One difference is that all the host-based B4s will use the same well-known IPv4 address 192.0.0.2. To uniquely identify the host-based B4, the AFTR will use the host-based B4's IPv6 address, which is unique for the host.

Softwire-Id/IPv4/Prot/Port	IPv4/Prot/Port
2001:db8:0:1::1/192.0.0.2/TCP/10000	192.0.2.1/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 a unique Softwire-Id. The source IPv4 address is one of the well-known IPv4 addresses. 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.

Authors' Addresses

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

EMail: adurand@juniper.net

Ralph Droms
Cisco
1414 Massachusetts Avenue
Boxborough, MA 01714
USA

EMail: rdroms@cisco.com

James Woodyatt
Apple
1 Infinite Loop
Cupertino, CA 95014
USA

EMail: jhw@apple.com

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

EMail: yiu_lee@cable.comcast.com