

Technology Overview

Configuring Dual-Stack Lite for IPv6 Access

Release
11.2



Published: 2011-05-17

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Technology Overview Configuring Dual-Stack Lite for IPv6 Access

Release 11.2

Copyright © 2011, Juniper Networks, Inc.

All rights reserved.

Revision History

April 2011—R1 Junos OS 11.2

The information in this document is current as of the date listed in the revision history.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

- a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.
- b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.
- c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.
- d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.
- e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the

Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14 (ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at <http://www.gnu.org/licenses/gpl.html>, and a copy of the LGPL at <http://www.gnu.org/licenses/lgpl.html>.

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Table of Contents

Overview of Dual-Stack Lite	1
DS-Lite Implementation	2
Example: Configuring Dual-Stack Lite for IPv6 Access	5

Overview of Dual-Stack Lite

This document describes Dual-Stack Lite (DS-Lite), a technology that enables Internet service providers to move to an IPv6 network while simultaneously handling IPv4 address depletion.

Because IPv4 addresses are becoming depleted, broadband service providers (DSL, cable, and mobile) need new addresses to supply new customers. Providing IPv6 addresses alone is often not workable because most of the systems that make up the public Internet are still enabled to support only IPv4, and many customer systems do not yet fully support IPv6.

DS-Lite provides one solution to this problem for service providers. DS-Lite allows the service provider to migrate to an IPv6 access network without changing end-user software. The device that accesses the Internet remains the same.

The DS-Lite architecture uses IPv6-only links between the provider and the customer while maintaining the IPv4 (or dual-stack) hosts in the customer network.

When a customer's device sends an IPv4 packet to an external destination, DS-Lite encapsulates the IPv4 packet in an IPv6 packet for transport into the provider network. These IPv4-in-IPv6 tunnels are called *softwires*. Tunneling IPv4 over IPv6 is simpler than translation and eliminates performance and redundancy concerns.

The softwires terminate in a softwire concentrator at some point in the service provider network, which decapsulates the IPv4 packets and sends them through a carrier-grade Network Address Translation (NAT) device. There, the packets undergo source-NAT processing to hide the original source address.

IPv6 packets originated by hosts in the subscriber's home network are transported natively over the access network.

The IPv4 packets originated by the end hosts have private (and possibly overlapping) IP addresses. Therefore, NAT must be applied to these packets. If end hosts have overlapping addresses, Network Address Port Translation (NAPT) is needed.

Using NAPT, the system adds the source address of the encapsulating IPv6 packet in the subscriber network to the inside IPv4 source address and port. Because each customer's IPv6 address is unique, the combination of the IPv6 source address with the IPv4 source address and port creates an unambiguous mapping.

The system takes the following actions when it receives a responding IPv4 packet from outside the subscriber network:

- Matches the IPv4 destination address and port for the packet to a specific customer based on the IPv6 address in the mapping table
- Maps the packet's IPv4 destination address and port to the IPv4 destination address and port inside the subscriber network

- Encapsulates the IPv4 packet in an IPv6 packet using the mapped IPv6 address as the IPv6 destination address
- Forwards the packet to the customer

For more information, see the following documents:

- draft-ietf-softwire-dual-stack-lite-06, *Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion*, August 2010.
- RFC 2473, *Generic Packet Tunneling in IPv6 Specification*, December 1998.
- RFC 2663, *IP Network Address Translator (NAT) Terminology and Considerations*, August 1999.
- RFC 4787, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP, BCP 127*, January 2007.
- RFC 4925, *Softwire Problem Statement*, July 2007.
- RFC 5382, *NAT Behavioral Requirements for TCP, BCP 142*, October 2008.
- RFC 5508, *NAT Behavioral Requirements for ICMP, BCP 148*, April 2009.
- <http://www.potaroo.net/tools/ipv4/index.html>
- <http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml>

DS-Lite Implementation

Beginning with Junos OS Release 10.4, Juniper Networks has implemented an Address Family Transition Router (AFTR) in its Services Physical Interface Cards (PICs) and Services Dense Port Concentrators (DPCs). An AFTR consists of the combination of an IPv4-in-IPv6 tunnel end-point and an IPv4-IPv4 NAT implemented on the same device.

A Basic Bridging BroadBand Element (B4 or software initiator) 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. IPv6 packets coming from a B4 are sent to a Services PIC, where the system creates a softwire according to the configuration. The system then extracts the IPv4 packets, performs NAT rule lookup and address translation, and sends the translated IPv4 packets to the Internet. The system performs these functions in a single pass through the Services PIC.

In the reverse path, the system sends IPv4 packets to the Services PIC, where the reverse translation is performed. The resulting packet is encapsulated in an IPv6 packet corresponding to the proper softwire and sent to the B4.

The system automatically creates softwires as IPv6 packets are received. IPv4 flows created by the encapsulated packets are associated with the specific softwire that initially carried them. When the last IPv4 flow associated with a softwire is completed, the softwire itself goes away. Thus, there is no need to create or manage tunnel interfaces, which simplifies the configuration.

The number of established softwires does not affect throughput, and scalability is independent of the number of interfaces.

- Related Documentation**
- Example: Configuring Dual-Stack Lite for IPv6 Access on page 5
 - Stateful NAT64 Overview

Example: Configuring Dual-Stack Lite for IPv6 Access

This example contains the following sections:

- Requirements on page 5
- Configuration Overview and Topology on page 5
- Configuration on page 6

Requirements

The following hardware components can perform DS-Lite:

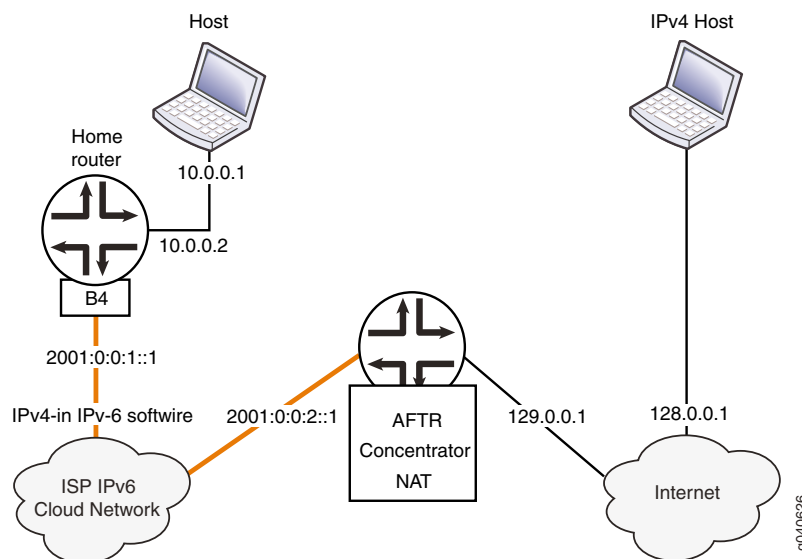
- M Series Multiservice Edge routers with Multiservices PICs
- T Series Core routers with Multiservices PICs
- MX Series 3D Universal Edge routers with Multiservices DPCs

The reference platform for this example is the MX Series router.

Configuration Overview and Topology

In Figure 1 on page 5, the AFTR is running on an MX Series router with two Gigabit Ethernet interfaces and a Multiservices DPC. The interface toward the Basic Bridging BroadBand Element (B4) is **ge-1/3/5**, and the interface toward the Internet is **ge-1/3/6**.

Figure 1: Logical Topology



In Figure 1 on page 5:

- The source IPv4 address connected to the home router is 10.0.0.1.
- The source address (or B4 interface address) of the IPv4-in-IPv6 software is 2001:0:0:1::1.
- The address of the NAT pool between the AFTR and the Internet is 129.0.0.1.

- The address of the IPv4 host connected to the Internet is 128.0.0.1.
- The address of the softwire on the AFTR is 2001:0:0:2::1/48.

Configuration

To configure DS-Lite involves the following tasks:

- Configuring the PIC in the ISP Network on page 6
- Configuring Interfaces and Service Sets on page 6
- Configuring a Service Set with Softwire and NAT Rules on page 7
- Configuring the Softwire Concentrator on page 9
- Anchoring the Softwire Concentrator on a Services PIC on page 10
- Using Show Commands to Verify DS-Lite Operation on page 10

Configuring the PIC in the ISP Network

Step-by-Step Procedure

1. Edit the **chassis** configuration to enable a Layer 3 service package. (The service package with its associated **sp-** interface is for manipulating traffic before it is delivered to its destination. For details about configuring service packages, see the *Junos OS Services Interfaces Configuration Guide*.)
2. Configure the service package at the **[edit chassis fpc pic adaptive-services]** hierarchy level. This example assumes that the PIC is in FPC 2, slot 0.

```
[edit chassis]
fpc 2 {
  pic 0 {
    adaptive-services {
      service-package layer-3;
    }
  }
}
```

Configuring Interfaces and Service Sets

Step-by-Step Procedure

1. Configure the **ge-1/3/5** interface between the home router running the B4 and the router in the ISP network running the AFTR.
 - a. Include the **family inet** (IPv4) and **family inet6** (IPv6) statements at the **[edit interfaces unit unit-number]** hierarchy level.
 - b. Include the IPv6 address of the AFTR router at the **[edit interfaces unit unit-number family inet6]** hierarchy level.
2. Configure a service set for the NAT and DS-Lite services at the **[edit interfaces interface-name unit unit-number family inet6 service input service-set]** and the **[edit interfaces unit unit-number family inet6 service output service-set]** hierarchy levels. This service set will be configured in a later step to include the softwire and NAT rules.
3. Include the address of the softwire at the **[edit interfaces interface-name unit unit-number family inet6 service address]** hierarchy level.

```
[edit interfaces]
ge-1/3/5 {
  description "AFTR-B4";
  unit 0 {
    family inet;
    family inet6 {
      service {
        input {
          service-set sset2;
        }
        output {
          service-set sset2;
        }
      }
    }
    address 2001:0:0:2::1/48;
  }
}
```

4. Configure the **ge-1/3/6** interface between the AFTR and the Internet.
 - a. Include the **family inet** statement at the **[edit interfaces interface-name unit unit-number]** hierarchy level.
 - b. Include the IPv4 address connected to the Internet at the **[edit interfaces interface-name unit unit-number family inet address]** hierarchy level.

```
[edit interfaces]
ge-1/3/6 {
  description "AFTR-Internet";
  unit 0 {
    family inet {
      address 128.0.0.1/24;
    }
  }
}
```

Configuring a Service Set with Software and NAT Rules

Step-by-Step Procedure To configure the service set on service interface **sp-2/0/0** to contain the software and NAT rules:

1. Configure a system log at the **[edit services service-set service-set-name]** hierarchy level.


```
[edit services service-set sset2]
syslog {
  host local {
    services any;
  }
}
```
2. Configure the service interface, in this example, **sp-2/0/0**.
 - a. Include the **family inet** and **family inet6** statements at the **[edit interfaces interface-name unit unit-number]** hierarchy level.

- b. Specify both the IPv4 and IPv6 address families in the **[edit interfaces interface-name unit unit-number]** hierarchy. The service set you configure in a later step is associated with this interface.

```
[edit interfaces]
sp-2/0/0 {
  unit 0 {
    family inet;
    family inet6 ;
  }
}
```

3. Because this release does not support fragmentation and reassembly, configure the maximum transmission units (MTUs) on the IPv6 and IPV4 networks.

Alternatively, you can configure TCP maximum segment size adjustment (TCP-MSS) to ensure that TCP traffic works through links with different MTUs.

This example configures a maximum segment size adjustment of 1024 at the **[edit services service-set service-set-name]** hierarchy level.

```
tcp-mss 1024;
```

4. Configure the NAT pool to specify the IPv4-to-IPv6 translation for packets traveling between the AFTR router and the Internet.
5. Configure an IPv4 address and port for the pool at the **[edit services nat pool pool-name]** hierarchy level.

```
[edit services nat]
pool p1 {
  address 129.0.0.1/32;
  port automatic;
}
```

6. Configure a NAT rule to translate the private IPv4 address from the home network to NAT **pool p1**. NAT rules specify the traffic to be matched and the action to be taken when traffic matches the rule. In this example, only one rule is required to accomplish the address translation. The rule selects all traffic coming from the source address 10.0.0.1.

```
[edit services]
rule r1 {
  match-direction input;
  term t1 {
    from {
      source-address {
        10.0.0.1/32;
      }
    }
    then {
      translated {
        source-pool p1;
        translation-type {
          source dynamic;
        }
      }
    }
  }
}
```



```

        syslog;
    }
}

```

7. Associate the software and NAT rules (this example uses the same rule for both) and the service interface with the service set at the **[edit services service-set service-set-name]** hierarchy level.

```

[edit services service-set]
software-rules r1;
nat-rules r1;
interface-service {
    service-interface sp-2/0/0.0;
}

```

Configuring the Software Concentrator

Step-by-Step Procedure

1. Create a software concentrator object of type **ds-lite** and associate it with the IPv6 address of the software. Give the software concentrator a name to facilitate references in logs, in the CLI, and in other operations and management activities. In this example, the IPv6 addresses of the interface facing the B4 and the software concentrator are the same.
2. As part of the software configuration, create a software rule. The rule in this example specifies that any traffic destined for the software concentrator **ds1** will create a new software. You can also configure more elaborate match conditions to perform as part of software initiator actions.

```

[edit services software]
software-concentrator {
    ds-lite ds1 {
        software-address 2001:0:0:2::1;
    }
}
rule r1 {
    match-direction input;
    term t1 {
        then {
            ds-lite ds1;
        }
    }
}

```

3. After completing the configuration, commit the configuration on each router.

```

user@host> commit check
configuration check succeeds
user@host> commit

```

Anchoring the Softwire Concentrator on a Services PIC

Step-by-Step Procedure “Configuring the Softwire Concentrator” on page 9 used the same IPv6 address in both the access interface and softwire concentrator for simplification. You can also use a different IPv6 address for the softwire concentrator that is decoupled from and can be reached through any access interface.

1. Change the address of the softwire concentrator to anchor the concentrator on an IPv6 address independent of any interface and on a different prefix.

```
[edit services softwire]
software-concentrator {
  ds-lite ds1 {
    software-address 2002::2:0:0:1;
  }
}
```

Using Show Commands to Verify DS-Lite Operation

Step-by-Step Procedure 1. On the host router, use the **show services stateful-firewall flows** command to verify the creation of the softwires, pre-NAT flows, and post NAT flows within the configuration.

```
user@host> show services stateful-firewall flows
```

```
Interface: sp-2/0/0, Service set: sset2
Flow
TCP      128.0.0.1:80  ->  129.0.0.2:1025      State   Dir   Frm count
          NAT dest    129.0.0.2:1025      ->  10.0.0.1:1025      Forward  O     2
          Softwire    2001:0:0:1::1       ->  2001:0:0:2::1
TCP      10.0.0.1:1025 ->  128.0.0.1:80         State   Dir   Frm count
          NAT source   10.0.0.1:1025       ->  129.0.0.2:1025      Forward  I     4
          Softwire    2001:0:0:1::1       ->  2001:0:0:2::1
```

In this example:

- In the output direction (O), the protocol (TCP) line shows the Internet-to-IPv4 host address translated to the address of the AFTR.
 - In the output direction, the NAT-translated IPv4 address is translated to the IPv4 address of the home host (NAT dest).
 - In the output direction, the IPv6 address of the B4 is translated to the IPv6 address of the AFTR (Softwire).
 - In the input direction (I), the protocol (TCP) line shows the address of the home host sending the packet to the address of the Internet-to-IPv4 host.
 - The input direction also shows the IPv6 address of the B4 being translated to the IPv6 address of the AFTR (NAT source).
2. Use the **show services stateful-firewall conversations** command to verify the conversations (collections of related flows). For example:

```
user@host> show services stateful-firewall conversations
```

Interface: sp-2/0/0, Service set: sset2

Conversation: ALG protocol: tcp

Number of initiators: 1, Number of responders: 1

Flow	Source		State	Dir	Frm count
TCP	10.0.0.1:1133	-> 128.0.0.1:80	Forward	I	7
NAT source	10.0.0.1:1133	-> 129.0.0.1:1425			
Software	2001:0:0:1::1	-> 2001:0:0:2::1			
TCP	128.0.0.1:80	-> 129.0.0.1:1425	Forward	O	5
NAT dest	129.0.0.1:1425	-> 10.0.0.1:1133			
Software	2001:0:0:1::1	-> 2001:0:0:2::1			

Conversation: ALG protocol: tcp

Number of initiators: 1, Number of responders: 1

Flow	Source		State	Dir	Frm count
TCP	10.0.0.1:1132	-> 128.0.0.1:80	Forward	I	6
NAT source	10.0.0.1:1132	-> 129.0.0.1:1424			
Software	2001:0:0:1::1	-> 2001:0:0:2::1			
TCP	128.0.0.1:80	-> 129.0.0.1:1425	Forward	O	5
NAT dest	129.0.0.1:1424	-> 10.0.0.1:1133			
Software	2001:0:0:1::1	-> 2001:0:0:2::1			

Conversation: ALG protocol: tcp

Number of initiators: 1, Number of responders: 1

Flow	Source		State	Dir	Frm count
TCP	10.0.0.1:1134	-> 128.0.0.1:80	Forward	I	7
NAT source	10.0.0.1:1134	-> 129.0.0.1:1426			
Software	2001:0:0:1::1	-> 2001:0:0:2::1			
TCP	128.0.0.1:80	-> 129.0.0.1:1425	Forward	O	5
NAT dest	129.0.0.1:1424	-> 10.0.0.1:1134			
Software	2001:0:0:1::1	-> 2001:0:0:2::1			

3. Use the **show services nat pool detail** command to display global NAT statistics related to pool usage. (You normally use this command in conjunction with the **show services stateful-firewall flows** command, which displays the source and output of the translation.) For example:

```
user@host> show services nat pool detail
```

Interface: sp-2/0/0, Service set: sset2

NAT pool: p1, Translation type: dynamic

Address range: 129.0.0.1-129.0.0.1

Port range: 512-65535, Ports in use: 16, Out of port errors: 0, Max ports used: 17

4. Examine the traceroute. The following output of a traceroute from the client, to the home host, to the IPv4 host on the Internet is based on "Configuring the Software Concentrator" on page 9.

```
user@host> show services stateful-firewall flows
```

Interface: sp-5/0/0, Service set: sset2

Flow	Source		State	Dir	Frm count
ICMP	10.0.0.1	-> 128.0.0.1	Watch	I	4
NAT source	10.0.0.1	-> 129.0.0.1			
Software	2001:0:0:1::1	-> 2002:0:0:2::1			
IPIP	2001:0:0:1::1:0	-> 2002:0:0:2::1:0	Forward	I	0
Software	2001:0:0:1::1	-> 2002:0:0:2::1			
ICMP	128.0.0.1	-> 129.0.0.1	Watch	O	1
NAT dest	129.0.0.1	-> 10.0.0.1			
Software	2001:0:0:1::1	-> 2002:0:0:2::1			



NOTE: If a traceroute starts from the home host and goes to an IPv4 host on the Internet, the software concentrator does not return an ICMP error and, therefore, is not properly identified as an intermediate hop. However, the traceroute still functions.

- Related Documentation**
- Overview of Dual-Stack Lite on page 1
 - Example: Configuring Stateful NAT64 for Handling IPv4 Address Depletion