JUNIPER NETWORKS

Technology Overview

Configuring Dual-Stack Lite for IPv6 Access

Release
11.3

Published: 2011-09-22

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

Technology Overview Configuring Dual-Stack Lite for IPv6 Access
Release 11.3

Revision History
September 2011—R1 Junos OS 11.3

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

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

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 softwire 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

The following hardware components can perform DS-Lite:

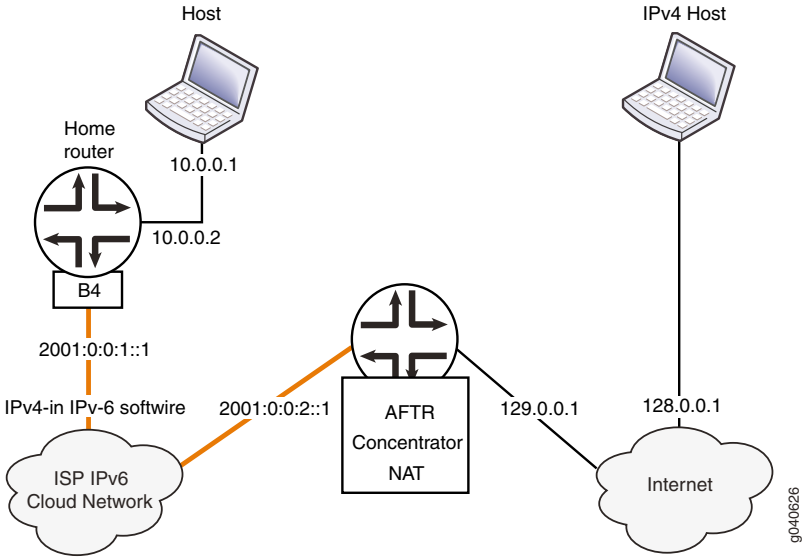
- M Series Multiservice Edge routers with Multiservices PICs
- T Series Core routers with Multiservices PICs
- MX Series 3D Universal Edge routers with Multiservices DPCs

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 softwire 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

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 Softwire and NAT Rules

Step-by-Step Procedure

To configure the service set on service interface **sp-2/0/0** to contain the softwire 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 softwire 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]
softwire-rules r1;
nat-rules r1;
interface-service {
  service-interface sp-2/0/0.0;
}
```

Configuring the Softwire Concentrator

Step-by-Step Procedure

1. Create a softwire concentrator object of type **ds-lite** and associate it with the IPv6 address of the softwire. Give the softwire 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 softwire concentrator are the same.

2. As part of the softwire configuration, create a softwire rule. The rule in this example specifies that any traffic destined for the softwire concentrator **ds1** will create a new softwire. You can also configure more elaborate match conditions to perform as part of softwire initiator actions.

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

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                                                 State      Dir     Frm count
TCP     128.0.0.1:80   ->    129.0.0.2:1025           Forward      O         2
  NAT dest    129.0.0.2:1025     ->    10.0.0.1:1025
  Softwire     2001:0:0:1::1     ->    2001:0:0:2::1
TCP      10.0.0.1:1025     ->   128.0.0.1:80          Forward      I         4
  NAT source    10.0.0.1:1025       ->    129.0.0.2:1025
  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
  Softwire     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
  Softwire    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
  Softwire     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
  Softwire    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
  Softwire     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
  Softwire    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 Softwire Concentrator" on page 9.

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

Interface: sp-5/0/0, Service set: sset2
Flow                                                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
     Softwire     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
     Softwire    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
     Softwire    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 softwire 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