Technical Guide

IPv6 Tunneling

Feature Overview and Configuration Guide

Introduction

This guide describes IPv6 Tunneling and its configuration, as specified in RFC2473.

IPv6 Tunneling is a mechanism for encapsulating IPv4 and IPv6 packets inside IPv6 packets.

Contents

Products and software version that apply to this guide

This guide applies to AlliedWare Plus™ products that support IPv6 Tunneling, running version **5.4.8-2.1** or later.

To see whether your product supports IPv6 Tunneling, see the following documents:

- The product's Datasheet
- The product's Command Reference

These documents are available from the above links on our website at alliedtelesis.com.

What is IPv6 Tunneling?

IPv6 Tunneling is a mechanism for encapsulating IPv4 and IPv6 packets inside IPv6 packets. It is used to form a virtual point-to-point link between two IPv6 nodes.

IPv6 Tunnels are stateless and have no knowledge of the configuration or even existence of the remote tunnel endpoint. Once an IPv6 Tunnel is configured, packets are encapsulated and forwarded regardless of whether the decapsulating device is present or not.

IPv6 Tunneling allows hosts in one private IP network to communicate with hosts in another private IP network by providing a tunnel between two routers across the Internet.

The IPv6 tunnel connection endpoints are terminated via a Virtual Tunnel Interface (VTI) configured in each device.

Figure 1: An IPv6 tunnel encapsulated packet form

Virtual Tunnel Interface (VTI)

A Virtual Tunnel Interface has similar characteristics to any other interface on the device. It is virtual because it does not directly map to any of the physical interfaces on the device, but instead is actually the endpoint of a tunnel from another device. VTIs are commonly layer 3 interfaces, can have IP configuration applied directly to them and are compatible with layer 3 routing protocols. The actual tunneling mechanism depends on the protocol used (GRE, RFC2473, L2TP and so on), but commonly uses IP as its transport.

Tunnel header

This is the outer or encapsulating IPv6 header. IPv6 Tunneling uses a standard IPv6 outer header and can be followed by extension headers as specified in IPv6 standards.

Payload header

This is the inner or encapsulated header. IPv6 Tunnels can be used to transport IPv4 and IPv6 packets.

IP packets from the private IP network destined for a host in the private IP network are encapsulated by Router A and forwarded to Router B. Intermediate routers route the packets using addresses in the delivery protocol header. Router B extracts the original payload packet and routes it to the appropriate destination within network.

The device supports the following features:

- IPv6 Tunneling as specified in RFC2473
- Virtual Tunnel Interfaces for terminating IPv6 encapsulated traffic
- IPv6 as the delivery protocol, used to transport the private data across the public network
- IPv4 as the payload, including DHCP, DNS
- IPv6 as the payload, including PIM6
- Configurable tunnel source using IPv6 address
- Configurable tunnel source using interface
- Configurable tunnel destination IPv6 address
- Configurable tunnel destination using hostname
- Configurable hop limit TTL value for insertion into the outer header
- Configurable DSCP value for insertion into the outer header
- Display of tunnel parameters in show interface output
- Tunnels are compatible with dynamic routing protocols
- Inherit the Flow Label from the inner header
- Insert Encapsulation Limit Extension Header if inner packet contains that header
- Path-MTU-discovery in the underlying tunnel interface
- TCP MSS Clamping

The device does not support the following features:

- Non-IPv4/IPv6 protocol as the payload
- Configurable Flow Label insertion
- IP Security tunnel protection

Configuration Example

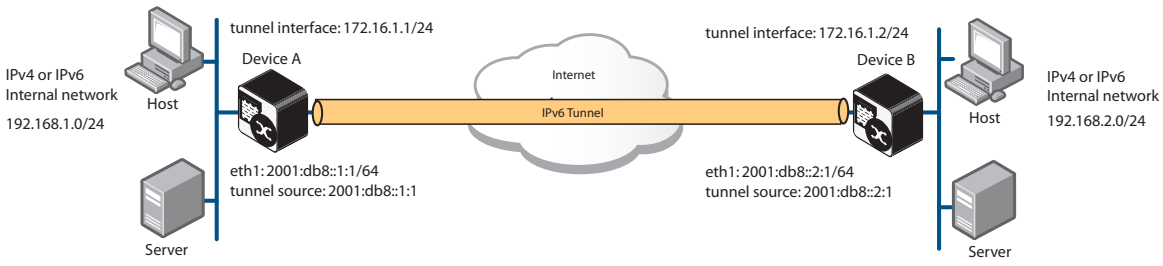
This example shows how to configure a IPv6 tunnel between Device A and Device B. It assumes that IP has been configured correctly and is operational on both devices.

The following table lists the parameter values in the example. Note public IP addresses are used in this example.

Table 1: Parameters in IPv6 Tunnel Configuration Example

PARAMETER	DEVICE A	DEVICE B
IP address of Ethernet interface eth1	**2001:db8::1:1/64**	**2001:db8::2:1/64**
Tunnel source IP address	**2001:db8:1::1**	**2001:db8::2:1**
Tunnel destination IP address	**2001:db8::2:1**	**2001:db8::1:1**
IP address of tunnel interface	**172.16.1.1/24**	**172.16.1.2/24**
Subnet of connected internal network	**192.168.1.0/24**	**192.168.2.0/24**

Figure 2: IPv6 Tunnel



Configuring device A

Step 1: **Assign an IP address for interface eth1.**

```
awplus# configure terminal
awplus(config)# interface eth1
awplus(config-if)# ipv6 address 2001:db8::1:1/64
```

Step 2: **Create tunnel interface tunnel1.**

```
awplus(config-if)# interface tunnel1
```

Step 3: **Assign an IP address to the tunnel interface.**

```
awplus(config-if)# ip address 172.16.1.1/24
```

Step 4: **Set the encapsulation tunneling mode to IPv6 Tunneling.**

```
awplus(config-if)# tunnel mode ipv6
```

Step 5: **Assign an IP address to the tunnel source for the tunnel.**

```
awplus(config-if)# tunnel source 2001:db8::1:1
```

Step 6: **Designate the tunnel destination address.**

```
awplus(config-if)# tunnel destination 2001:db8::2:1
```

Step 7: **Configure a static route.**

```
awplus(config-if)# exit
awplus(config)# ip route 192.168.2.0 255.255.255.0 172.16.1.2
```

Configuring device B

Step 1: **Assign an IP address for interface eth1.**

```
awplus# configure terminal
awplus(config)# interface eth1
awplus(config-if)# ipv6 address 2001:db8::2:1/64
```

Step 2: **Create tunnel interface tunnel1.**

```
awplus(config-if)# interface tunnel1
```

Step 3: **Assign an IP address to the tunnel interface.**

```
awplus(config-if)# ip address 172.16.1.2/24
```

Step 4: **Set the encapsulation tunneling mode to IPv6 Tunneling.**

```
awplus(config-if)# tunnel mode ipv6
```

Step 5: **Assign an IP address to the tunnel source for the tunnel.**

```
awplus(config-if)# tunnel source 2001:db8::2:1
```

Step 6: **Designate the tunnel destination address.**

```
awplus(config-if)# tunnel destination 2001:db8::1:1
```

Step 7: **Configure a static route.**

```
awplus(config-if)# exit
awplus(config)# ip route 192.168.1.0 255.255.255.0 172.16.1.1
```

Verifying connectivity

You can use the **ping** command to verify that the tunnel is established:

```
awplus# ping 192.168.2.1
```

You should receive ICMP Echo reply message.

Verifying the tunnel settings

You can use the **show interface tunnel** command to check that all settings are correctly configured:

```
awplus# show interface tunnel1
```

The output will show the settings for the tunnel.

Figure 3: Example output from the show interface tunnel command

```
awplus#show interface tunnel1
Interface tunnel1
  Link is UP, administrative state is UP
  Hardware is Tunnel
  IPv4 address 192.168.10.1/24 pointopoint 192.168.10.255
  index 4751 metric 1 mtu 1480
  arp ageing timeout 300
  <UP,POINTOPOINT,RUNNING,MULTICAST>
  SNMP link-status traps: Disabled
  Tunnel source 2001:db8::1:1, destination 2001:db8::2:1
  Tunnel name local 2001:db8::1:1, remote 2001:db8::2:1
  Tunnel ID local (not set), remote (not set)
  Tunnel protocol/transport ipv6, key disabled, sequencing disabled
  Tunnel TTL 64
  Checksumming of packets disabled, path MTU discovery disabled
    input packets 0, bytes 0, dropped 0, multicast packets 0
    output packets 0, bytes 0, multicast packets 0 broadcast packets 0
  Time since last state change: 0 days 22:38:35
```


Allied Telesis NETWORK SMARTER

North America Headquarters | 19800 North Creek Parkway | Suite 100 | Bothell | WA 98011 | USA | T: +1 800 424 4284 | F: +1 425 481 3895
Asia-Pacific Headquarters | 11 Tai Seng Link | Singapore | 534182 | T: +65 6383 3832 | F: +65 6383 3830
EMEA & CSA Operations | Incheonweg 7 | 1437 EK Rozenburg | The Netherlands | T: +31 20 7950020 | F: +31 20 7950021

alliedtelesis.com