Technical Guide

IPv6 over IPv4 Tunneling

Feature Overview and Configuration Guide

Introduction

This guide describes IPv6 over IPv4 tunneling and how to configure it. IPv6 over IPv4 tunnels are point-to-point tunnels made by encapsulating IPv6 packets within IPv4 headers to carry them over IPv4 routing infrastructures. This allows isolated IPv6 end systems and devices to communicate without the need to upgrade the IPv4 infrastructure that exists between them.

Moving a network from IPv4 addressing to IPv6 addressing cannot be performed in a single step. The transition necessarily proceeds in stages, with islands of IPv6 developing within the IPv4 network, and gradually growing until they cover the whole network. During this transition process, the islands of IPv6 need to be able to communicate with each other across the IPv4 network. Additionally, it is desirable to be able to transition some network functions across to IPv6 while the majority of the network is still using IPv4. During early transition, IPv4 networks are widely deployed and IPv6 networks are isolated sites. An IPv6 over IPv4 tunnel allows IPv6 packets to be transmitted on an IPv4 network and connects all IPv6 sites.

A variety of transition mechanisms are available for tunneling IPv6 over existing IPv4 networks:

- manually configured IPv6 over IPv4 tunneling
- IPv6 over IPv4 GRE tunneling
- semi-automatic tunneling
- fully automatic tunneling
- ISATAP tunneling

This guide describes **manually configured IPv6 over IPv4 tunneling** on AlliedWare Plus™ devices.

Contents

Products and software version that apply to this guide

This guide applies to AlliedWare Plus™ products that support IPv6 over IPv4 tunneling running version **5.5.0-0.1** or later.

To see whether your product supports the IPv6 over IPv4 tunneling mode, 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.

Related documents

The following documents give more information about the IPv6 transitioning and tunneling features on AlliedWare Plus products:

For AR-series firewalls and routers only:

- the Transitioning IPv4 to IPv6 Feature Overview Guide
- the IPv6 Tunneling Feature Overview Guide

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

Licensing

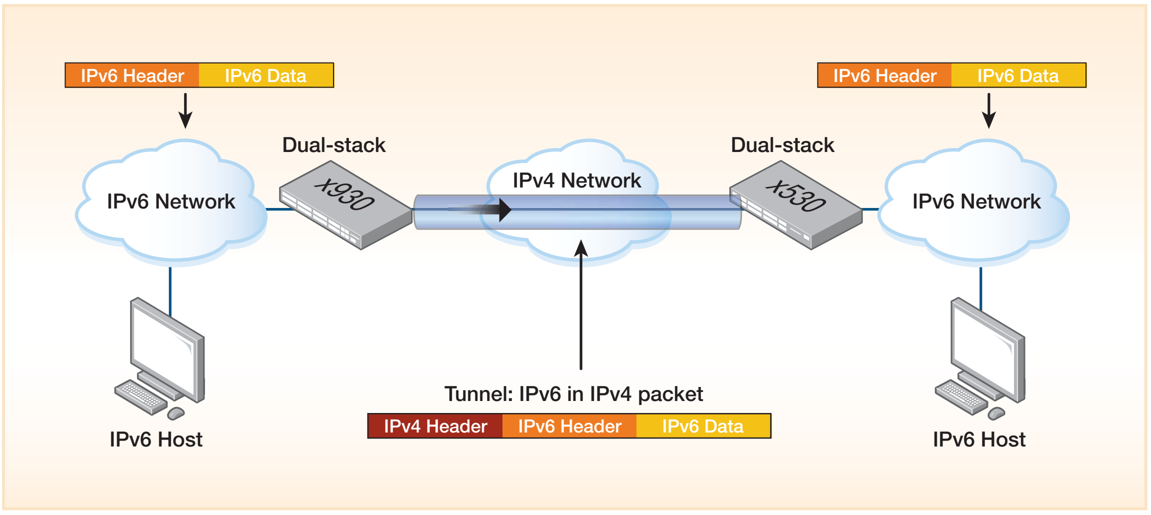
This feature is part of the standard feature set of the device software, therefore there are no licensing requirements.

Deploying IPv6 over IPv4 tunnels

The key to a successful IPv6 transition is compatibility with the large installed base of IPv4 hosts and routers. Maintaining compatibility with IPv4 while deploying IPv6 will streamline the task of transitioning the Internet to IPv6.

An IPv6 over IPv4 tunnel connects isolated IPv6 sites through an IPv4 network. Deploying IPv6 over IPv4 tunneling can be useful for service providers and enterprises who want to offer an end-to-end IPv6 service without major upgrades to the infrastructure. One of the major benefits of this type of tunneling is the ability to interconnect isolated IPv6 domains over existing IPv4 infrastructures.

Figure 1: An IPv6 over IPv4 tunnel



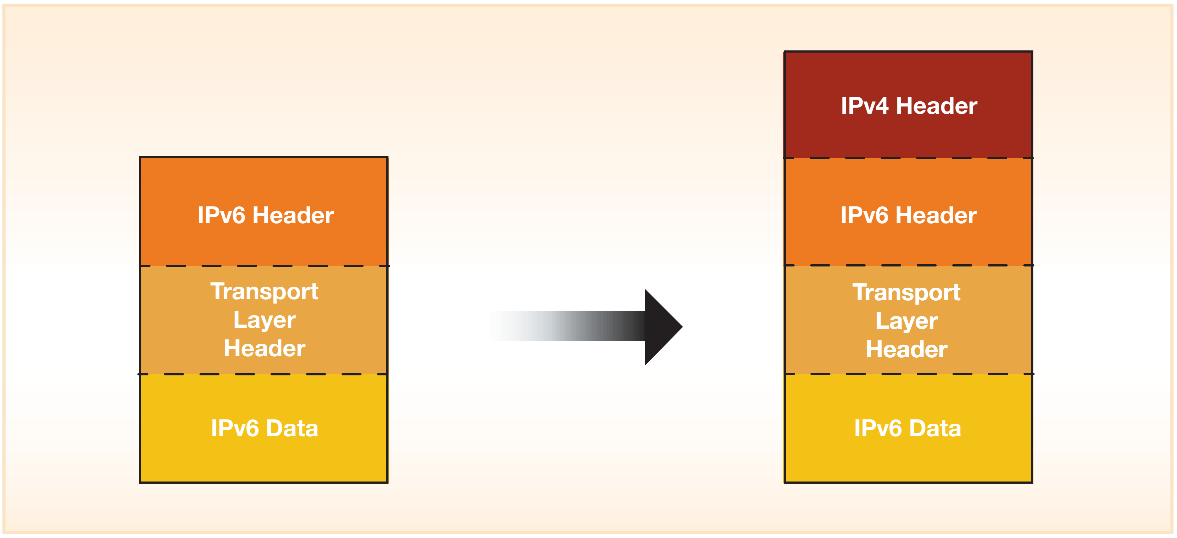
With IPv6 over IPv4 tunneling, the IPv4 tunnel endpoint address is determined by configuration information on the encapsulating node. The tunnels can be either unidirectional or bidirectional. Bidirectional configured tunnels behave as virtual point-to-point links.

As IPv4 and IPv6 networks are not directly inter-operable, there are transition mechanisms available that permit hosts on either network type to communicate with any other host. Transition mechanisms bridge between IPv4 and IPv6 and allow the two versions to work side by side. Because IPv6 is a completely separate protocol from IPv4, it can be run in parallel with IPv4 as the transition is made from IPv4 to IPv6. Hosts and network devices can run both IPv4 and IPv6 on the same interface at the same time (dual-stacked), and each is invisible to the other; there is no interference between the two protocols. In time, IPv4 networks will fade away as we move to the IPv6 addressing system.

Encapsulation

The encapsulation of an IPv6 datagram in IPv4 is shown below:

Figure 2: Encapsulating an IPv6 datagram inside IPv4



In addition to adding an IPv4 header, the encapsulating node also has to handle some more complex issues such as:

- Determining when to fragment and when to report an ICMP "packet too big" message back to the source.
- How to reflect IPv4 ICMP messages from routers along the tunnel path back to the source as IPv6 ICMP errors.

Tunneling requirements

All tunneling mechanisms require that the endpoints of the tunnel run both IPv4 and IPv6 protocol stacks, that is, endpoints must run in dual-stack mode. A dual stack network is a network in which all of the nodes are both IPv4 and IPv6 enabled.

Dual-stack devices run both IPv4 and IPv6 protocols simultaneously and thus can inter-operate directly with both IPv4 and IPv6 end systems and devices. For proper operation of the tunnel mechanisms, appropriate entries in a DNS that map between host names and IP addresses for both IPv4 and IPv6 allow the applications to choose the required address.

For more information on encapsulation and tunneling requirements, please refer to RFC4213.

Manually configured tunneling

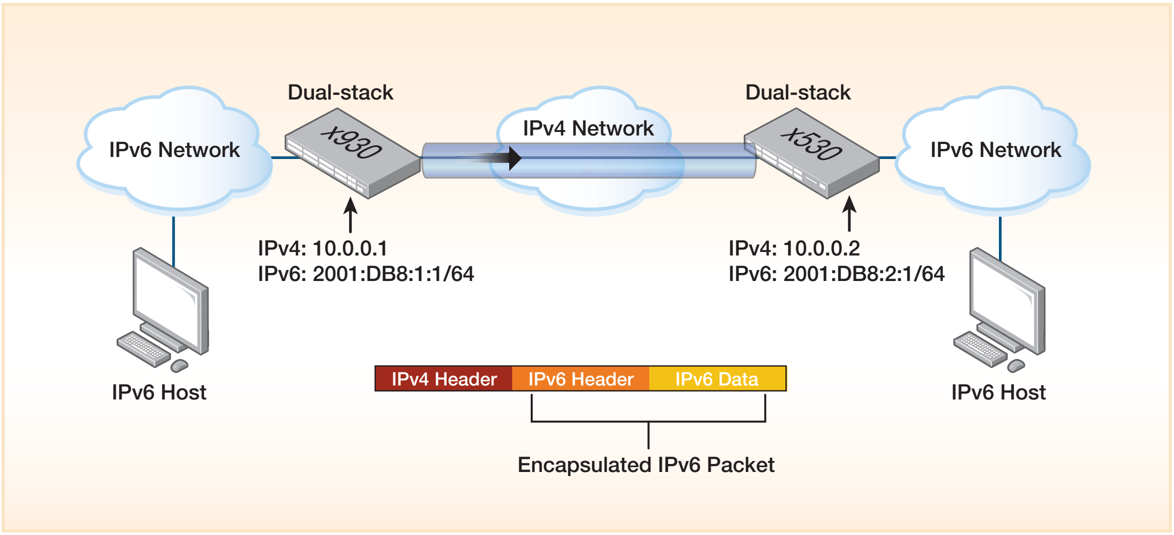
The primary use of a manually configured tunnel is to provide stable and secure connections for regular communication between two edge devices, or between an end system and an edge device, or for connection to remote IPv6 networks. The edge devices and end systems used as tunnel endpoints must be dual-stack devices. Manually configured tunnels are used between two points and require configuration of both the source and destination addresses of the tunnel, where as automatic tunnel mechanisms need to be only enabled and are more transient.

In manually configured tunneling, the tunnel endpoint address is determined from configuration information in the encapsulating node. For each tunnel, the encapsulating node must store the tunnel endpoint address.

Because each tunnel is independently managed, the more tunnel endpoints you have, the more tunnels you need, and the greater is the management overhead. As with other tunnel mechanisms, network address translation (NAT) is not allowed along the path of the tunnel. The configuration example belows provides the steps required to configure an IPv6 over IPv4 tunnel.

Configuration example

To create an IPv6 over IPv4 tunnel between two AlliedWare Plus devices, follow the example configuration steps provided below:



x930 Step 1: **Assign an IP address to the VLAN interface and enable IPv6**

```
awplus#configure terminal
awplus(config)#int vlan2
awplus(config-if)# ip address 10.0.0.1/27
awplus(config-if)# ipv6 enable
```

Step 2: Create the tunnel and set the tunnel mode

```
awplus(config)#int tunnel1
awplus(config-if)# tunnel source vlan2
awplus(config-if)# tunnel destination 10.0.0.2
awplus(config-if)# tunnel mode ipv6ip
awplus(config-if)# ipv6 address 2001:DB8:1:1/64
awplus(config-if)# ipv6 enable
```

Step 3: Configure a static route

```
awplus(config)#ipv6 forwarding
awplus(config)#ipv6 route 2001:DB8:2/48 tunnel1
```

Note: The static route must be configured for traffic matching subnet `2001:DB8:2/48` to be routed through the tunnel to the x530 switch

x530

Step 1: Assign an IP address to the VLAN interface and enable IPv6

```
awplus#configure terminal
awplus(config)#int vlan2
awplus(config-if)# ip address 10.0.0.2/27
awplus(config-if)# ipv6 enable
```

Step 2: Create the tunnel and set the tunnel mode

```
awplus(config)#int tunnel1
awplus(config-if)# tunnel source vlan2
awplus(config-if)# tunnel destination 10.0.0.1
awplus(config-if)# tunnel mode ipv6ip
awplus(config-if)# ipv6 address 2001:DB8:2:1/64
awplus(config-if)# ipv6 enable
```

Step 3: Configure a static route

```
awplus(config)#ipv6 forwarding
awplus(config)#ipv6 route 2001:DB8:1/48 tunnel1
```

Troubleshooting

There are a number of show commands available to help troubleshoot IPv6 over IPv4 tunneling.

Use the following show commands to troubleshoot the tunnels:

- show platform mem tunnel (x530)
- show platform table tunnel (x930)
- show platform table tunnelterm (x930)

It’s always good to know what to look for in any show output. For all the above show commands, the important thing is that the Source IP interface and Destination IP are the same with the tunnel interface configuration.

If the show command has no output, it is a clear indication that the tunnel is not working properly.

Figure 3: This incomplete show output indicates a tunnel issue

```
x930-STK#show platform table tunnel

Stack member 1:

[Instance 4]
Intf  Type  TTL  DF  DSCP    Source IP     Destination IP
---------------------------------------------------------

Stack member 2:

[Instance 8]
Intf  Type  TTL  DF  DSCP     Source IP    Destination IP
---------------------------------------------------------
```

If the tunnel is working correctly, you should expect to see output similar to that shown below for the command **show platform table tunnel**:

Figure 4: This output indicates no tunnel issue

```
interface vlan10
ip address 10.1.1.1/24
ipv6 enable
!
interface vlan100
ipv6 address 2001:db8:100::1/64
ipv6 enable
!
interface tunnel1
tunnel source vlan10
tunnel destination 20.1.1.2
tunnel mode ipv6ip
ipv6 address 2001:db8:1020::1/64
ipv6 enable
!
DUT_x930-STK#show platform table tunnel

Stack member 1:

[Instance 4]
Intf  Type  TTL  DF  DCSP      Source IP    Destination IP
----------------------------------------------------------
3     3     255  0   Asgn 0    10.1.1.1     20.1.1.2

Stack member 2:

[Instance 8]
Intf  Type  TTL  DF  DCSP      Source IP    Destination IP
----------------------------------------------------------
3     3     255  0   Asgn 0    10.1.1.1     20.1.1.2
4     3     255  0   Asgn 0    10.1.1.1     20.1.1.2
5     3     255  0   Asgn 0    10.1.1.1     20.1.1.2
6     3     255  0   Asgn 0    10.1.1.1     20.1.1.2
7     3     255  0   Asgn 0    10.1.1.1     20.1.1.2
DUT_x930-STK#
```