Technical Guide

# Internet Protocol v6 (IPv6)

## Feature Overview and Configuration Guide

## Introduction

This guide describes the main features of IPv6, the implementation of IPv6 and how to configure and operate IPv6 on the device.

The following IPv6 features are discussed:

- IPv6 address and prefixes
- Neighbor and router discovery
- Stateless Address Autoconfiguration (SLAAC)
- RA Guard
- IPv6 Source Address Dependent Routing
- IPv6 neighbor discovery proxy for AR Series switches

### Products and software version that apply to this guide

This guide applies to AlliedWare Plus™ products that support IPv6, running version **5.4.4** or later.

The behavior of the command **ipv6 enable** is not applicable to software releases earlier than 5.4.7. See the section titled: "Configuring SLAAC" on page 16, for more detail.

Support and implementation of IPv6 varies between products. To see whether a product supports a particular feature or command, 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.

Feature support may change in later software versions. For the latest information, see the above documents.

# Content

# Overview

IPv6 is the next generation of the Internet Protocol (IP). It has primarily been developed to solve the problem of the eventual exhaustion of the IPv4 address space, but also offers other enhancements. IPv6 addresses are **16** bytes long, in contrast to IPv4's **4** byte addresses. Other features of IPv6 include:

**Address structure improvements**

- globally unique addresses with more levels of addressing hierarchy to reduce the size of routing tables
- improved scalability of **multicast** routing by adding a **scope** field to multicast addresses
- a new type of addressing method, the **anycast address**, which sends packets to any one of a group of devices. This method removes the need for packet fragmentation en-route, by dynamic determination of the largest packet size that is supported by every link in the path. A link's MTU (Maximum Transmission Unit) must be at least 1280 bytes, compared with 576 bytes for IPv4.

# IPv6 addresses and prefixes

IPv6 addresses have a hexadecimal format that are made up of eight pairs of octets separated by colons. An example of a valid address is **2001:0db8:0000:0000:0260:0000:97ff:64aa**. In the interests of brevity, addresses can be abbreviated in two ways:

- Leading zeros can be omitted, so this address can be written as: **2001:db8:0:0:260:0:97ff:64aa**.
- Consecutive zeros can be replaced with a double colon, so this address can be written as **2001:db8::260:0:97ff:64a**. Note that a double colon can replace any number of consecutive zeros, but an address can contain only one double colon.

Like IPv4 addresses, a proportion of the leftmost bits of the IPv6 address can be used to indicate the subnet, rather than a single node. This part of the address is called the ***prefix***. Prefixes provide the equivalent functionality to a subnet mask in IPv4, allowing a subnet to be addressed, rather than a single node. If a prefix is specified, the IPv6 address is followed by a slash and the number of bits that represent the prefix. For example, **2001::/16** indicates that the first 16 bits (**2001**) of the address **2001:0:0:0:0:0:0:0** represent the prefix.

Also like IPv4 addresses, IPv6 addresses are attached to interfaces. Note that IPv6 addressing is supported on PPP interfaces as well as VLAN, tunnel, and Ethernet interfaces.

## Address types

IPv6 supports the following address types:

- Unicast
- Multicast
- Anycast

### Unicast addresses

A unicast address is attached to a single interface and delivers packets only to that interface.

Unicast addresses can be grouped into 3 address scopes:

1. Link-local
2. Global
3. Unique-local

**Link-local**—these addresses start with **FE8x:** and are used in a single link or subnet. Any packets that are transmitted with a link local source/destination address scope are never routed out of that subnet.

**Global**—these addresses usually fall into the **2000::/3** prefix. They are the equivalent of public IPv4 addresses. Global scoped addresses can be routed publicly in the Internet. Any device or site that wishes to transmit packets to another site must be uniquely identified with a global address. Some global addresses are allocated to special purposes. FC00::/7 is outside the 2000::/3 prefix, but FC00::/7 is still considered as a global scope address.

**Unique-local**—are unique global scope addresses that cannot be routed across the global Internet IPv6 address space. Layer 3 devices will not forward any packets with unique-local source or destination addresses outside of the private enterprise or customer site. IPv6 routing between multiple unique-local subnets within a private enterprise is allowed.
There is a bit of history to which address ranges have become used for local addresses. Originally it was the range fec0: :/10 (RFC 1884). But the term ‘site-local’ was not well defined in the original definition of site-local addresses. The use of fec0: :/10 was deprecated in RFC 3879. Shortly later, a new range was defined - fc00: :/7 (RFC 4193) for Unique-local address ranges.

**Reserved for documentation**:

- 3FFF:FFFF::/32
- 2001:db8::/32

**Used for 6 to 4 tunneling**:

- 2002::/16

Note: The AlliedWare Plus implementation of 6 to 4 tunneling has been deactivated, although, IANA still allocated the 2002::/16 prefix as the 6 to 4 prefix. For information on the current special use addresses, see the: IPv6 Special-Purpose Address Registry.

**Used for IPv4 mapped IPv6 addresses**:

- ::ffff:0:0/96

  These addresses are typically written with a 96-bit prefix in the standard IPv6 format, and the remaining 32 bits written in the customary dot-decimal notation of IPv4. For example, ::ffff:198.0.2.128 represents the IPv4 address 198.0.2.128. A deprecated format for IPv4-mapped IPv6 addresses is ::198.0.2.128.

**The following special addresses have been defined**:

- The **loopback** address, consisting of **::1**, which is the equivalent of the IPv4 loopback address and allows a host to send packets to itself.
- The **unspecified** address, consisting of **::**, which is the equivalent of the IPv4 unspecified address and is used as a source address by hosts during the autoconfiguration process.

### Multicast addresses

Multicast addresses start with FFxx: and they operate the same as the IPv4 multicast addresses. Interfaces can belong to one or more multicast groups and will accept a multicast packet only if they belong to the group corresponding to the packet's destination address.

There are **no** broadcast packets in IPv6, instead the IPv6 protocol uses IPv6 multicast packets to do the job of an IPv4 broadcast packet. Multicasting provides a much more efficient mechanism than broadcasting, which requires that every host on a link accept and process each broadcast packet.

### Multicast address scopes

The scope of a multicast address is indicated by the fourth hex digit in the address, i.e. the digit 'E' in the FF0E:: prefix. The 4-bit scope field is used to indicate where the address is valid and unique. The following table lists some multicast address scopes:

Table 1: Multicast address scopes

| VALUE | SCOPE | MEANING |
|---|---|---|
| FF01:: | interface-local | The current scope as defined in RFC 7346 |
| FF02:: | link-local | Forwarded only within a subnet on an Ethernet segment |
| FF04:: | admin-local | Forwarded within a small administratively- defined topology |
| FF05:: | site-local | Forwarded only within a single site |
| FF08:: | organisational-local | Forwarding can span multiple sites of a single organization |
| FF0E:: | global | Can be sent across the Internet |

### Anycast addresses

An anycast address is a unicast address that is attached to more than one interface. If a packet is sent to an anycast address it is delivered to the nearest interface with that address, with the definition of "nearest" depending on the protocol used for routing. If the protocol is RIPv6, the nearest interface is the one that is the shortest number of hops away.

Anycast addresses packets cannot originate from an anycast address. An interface must be configured to know that it is using an anycast address because the address format cannot be distinguished from that of a unicast address.

Only one anycast address has been predefined: the subnet-router address. The subnet-router address sends messages to the nearest router on a subnet and consists of the subnet's prefix followed by zeros.

## IPv6 headers

The basic unit of IPv6 data sent through the Internet is called a packet. A packet consists of a header followed by the data. The following figure shows the IPv6 packet.

Figure 1: IPv6 packet structure

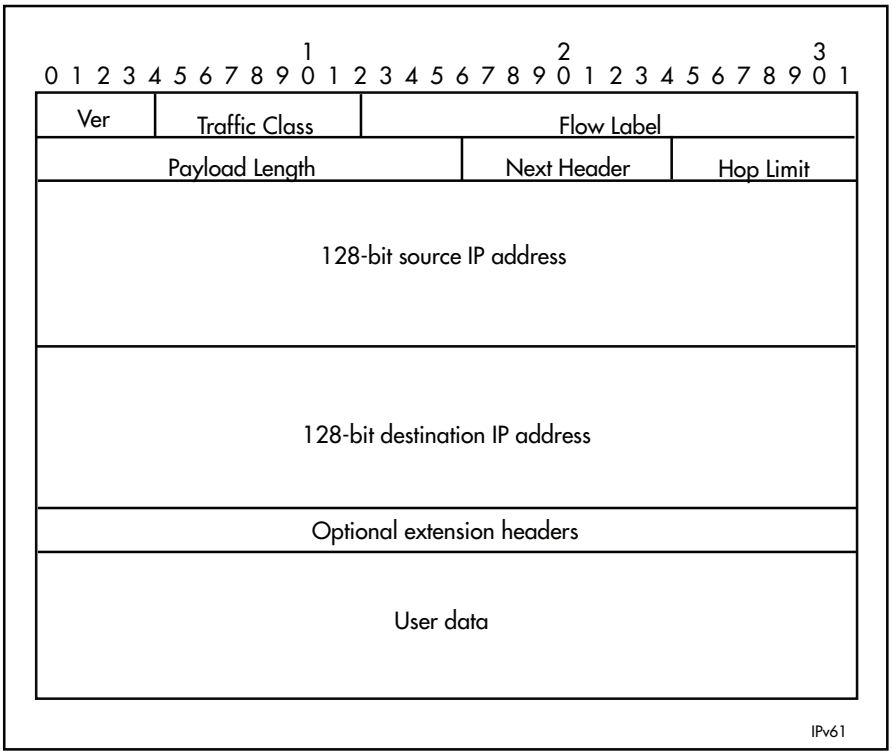



Table 2: IPv6 packet - Field descriptions

| FIELD | FUNCTION |
|---|---|
| Ver | Version of the IP protocol that created the packet. For IPv6, this field has a value of 6. |
| Traffic Class | 8-bit Traffic Class field that contains the 6-bit DSCP and is used to prioritize traffic as part of a Quality of Service system.<br>Traffic Class allows packets to be labelled with an appropriate priority. If the network becomes congested, the lowest priority packets are dropped<br>Additional information can be found in RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. |
| Flow Label | 20-bit value that indicates the data flow to which this packet belongs.This flow may be handled in a particular way.<br>Flow labels indicate to intermediate switches and routers that packets are part of a flow, and that a particular flow requires a particular type of service. This feature enables, for example, real-time processing of data streams. It also increases routing speed because the forwarding router or switch needs only to check the flow label, not the rest of the header. The handling indicated by the flow label can be done by the IPv6 Hop-by-Hop header, or by a separate protocol such as RSVP (Resource Reservation Protocol). |
| Payload Length | Length of the user data portion of the packet. If the data payload is larger than 64 kB, the length is given in the optional “Jumbo Payload” header and the Payload Length header is given a value of zero. |

Table 2: IPv6 packet - Field descriptions (Continued)

| FIELD | FUNCTION |
|---|---|
| Next Header | Number that indicates the type of header that immediately follows the basic IP header. This header type may be an optional IPv6 extension header, a relevant IPv4 option header, or another protocol, such as TCP or ICMPv6.<br>The IPv6 extension header values are:<br>**0** (Hop-by-Hop Options Header)<br>**43** (IPv6 Routing Header)<br>**44** (IPv6 Fragment Header)<br>**50** (Encapsulating Security Payload)<br>**51** (IPv6 Authentication Header)<br>**59** (No Next Header)<br>**60** (Destination Options Header) |
| Hop Limit | Field that is the equivalent of the IPv4 Time To Live field, measured in hops. |
| Source IP address | 128-bit IPv6 address of the sender. |
| Destination IP address | 128-bit IPv6 address of the recipient. |
| Optional extension headers | Headers for less-frequently used information. |
| User data | Payload. |

## Basic IPv6 header structure

The headers contain information necessary to move the packet across the Internet. They must be able to cope with missing and duplicated packets as well as possible fragmentation (and reassembly) of the original packet.IPv6 headers are twice as long as IPv4 headers (40 bytes instead of 20 bytes) and contain four times the address space size (128 bits instead of 32 bits).

They no longer contain the header length, identification, flags, fragment offset, and header checksum fields. Some of these options are placed in extension headers. The Time To Live field is replaced with a hop limit, and the IPv4 Type of Service field is replaced with the Traffic Class field.

The Differentiated Services field contains the DSCP bits, used in a Quality of Service (QoS) regime. RFC-8200 explains Traffic Class as:

- The 8-bit Traffic Class field in the IPv6 header is used by the network for traffic management. The value of the Traffic Class bits in a received packet or fragment might be different from the value sent by the packet's source.

The current use of the Traffic Class field for Differentiated Services and Explicit Congestion Notification is specified in [RFC2474] and [RFC3168].

The following table explains IPv4 header fields that changed in IPv6:

Table 3: IPv4 header fields changed in IPv6

| CHANGED FIELD | DESCRIPTION |
|---|---|
| Type of Service | The type of service that a connection should receive is indicated in IPv6 by the Flow Label field in the IPv6 header. |
| Fragmentation information (the Identification field, the Flags field and the Fragment Offset field) | In most cases fragmentation does not occur in IPv6. If it does, packets are fragmented at their source and not en route. Therefore, the fragmentation information is contained in an extension header to reduce the size of the basic IPv6 header. |
| Header Checksum | This option has not been provided in IPv6. This is because transport protocols implement checksums and because of the availability of the IPsec authentication header (AH) in IPv6. |
| Options | Extension headers handle all the optional values associated with IPv6 packets. The biggest advantage of this scheme is that the size of the basic IP header is a constant. |

## Extension headers

IPv6 implements many of the less commonly used fields in the IPv4 header (or their equivalents) as extension headers, which are placed after the basic IPv6 header. The length of each header must be a multiple of 8 bytes.

The first extension header is identified by the Next Header field in the basic IPv6 header. Any subsequent extension headers are identified by an 8-bit “Next Header” value at the beginning of the preceding extension header.

IPv6 nodes that originate packets are required to place extension headers in a specific order:

1. The basic IPv6 header. This must come immediately before the extension headers.
2. The Hop-by-Hop header. This specifies options that must be examined by every node in the routing path.
3. A Destination Options header. This is used to specify options to be processed by the first destination or final destination. The destination options header is the only extension header that may be present more than once in the IPv6 packet.
4. The Routing header. This enables a static path to be specified for the packet, if the dynamically-determined path is undesirable.
5. The Fragment header. This indicates that the source node has fragmented the packet, and contains information about the fragmentation.
6. The Authentication header (AH). This verifies the integrity of the packet and its headers.
7. The Encapsulating Security Payload (ESP) header. This encrypts a packet and verifies the integrity of its contents.
8. Another Destination Options Header. Processed by the final destination of the packet.
9. The Upper Layer Protocol header. This indicates which protocol a higher layer (such as the Transport layer) is to process the packet with (for example, TCP).

## The Internet Control Message Protocol (ICMPv6)

The Internet Control Message Protocol, ICMPv6, provides a mechanism for error reporting and route discovery and diagnostics. It also conveys information about multicast group membership, a function that is carried out by the Internet Group Management Protocol (IGMP) in IPv4, and performs address resolution, which the Address Resolution Protocol (ARP) performs in IPv4.

Significant aspects of ICMPv6 include neighbor discovery, which enables one device in a network to find out about other nearby devices; and stateless address autoconfiguration, which allows a device to dynamically determine its own IPv6 address.

ICMPv6 is also used to support the Ping v6 (Packet Internet Groper) and Trace route v6 functions that are used to verify the connections between networks and network devices. Ping is used to test the connectivity between two network devices to determine whether each network device can “see” the other device. Trace route is used to discover the route used to pass packets between two systems running the IP protocol.

Ping and Trace route operate almost identically in IPv4 and IPv6.

## Neighbor Discovery

Neighbor Discovery is an ICMPv6 function that enables a router or a host to identify other devices on its links.

The IPv6 Neighbor Discovery protocol is similar to a combination of the IPv4 protocols ARP, ICMP Router Discovery and ICMP Redirect.

The following table describes packet types involved with IPv6 Neighbor Discovery:

Table 4: Packet types involved with neighbor discovery

| PACKET TYPE | DESCRIPTION |
|---|---|
| router solicitation | Packet in which a host sends out a request for routers to generate advertisements. |
| router advertisement | Allows routers to advertise their presence and other network parameters. A router sends an advertisement packet in response to a solicitation packet from a host. |
| neighbor solicitation | Packet in which a node sends a packet to determine the link layer address of a neighbor or to verify that a neighbor is still active. |
| neighbor advertisement | A response to a neighbor solicitation packet. These packets are also used to notify neighbors of link layer address changes. |
| redirect | Informs hosts of a better first hop. |

Note: To comply with Section 6.2.1 of RFC 4861, IPv6 Neighbor Discovery, the router does not generate Router Advertisements by default.

The following table explains packet types and services:

Table 5: Packet types and services

| PACKET TYPE | DESCRIPTION |
|---|---|
| address resolution | Translation of Layer 3 destination to the Layer 2 address. This is achieved using the Neighbor Solicitation Message and the Neighbor Advertisement Message. |
| router and prefix discovery | On connection to a link, a node needs to know the address of a router that the node can use to reach the rest of the world. The node also needs to know the prefix (or prefixes) that define the range of IP addresses on its link that it can reach without going through a router.<br>Routers use ICMP to convey this information to hosts, by means of Router Advertisements. The message may have an option attached (the source link address option), which enables the receiving node to respond directly to the router, without performing a neighbor solicitation. |
| immediate information | The configuration of a router includes a defined frequency at which unsolicited advertisements are sent. If a node wants to obtain information about the nearest router immediately, rather than waiting for the next unsolicited advertisement, the node can send a router solicitation message. Each router that receives the solicitation message sends a Router Advertisement specifically to the node that sent the solicitation. |
| redirection | If a node is aware of more than one router that it can use to connect to wider networks, the router to which it sends packets by default does not always represent the most desirable route. Routers use the ICMPv6 redirect packet to communicate a more effective path to the node. |
| Neighbor Unreachability Detection (NUD) | A node may issue solicitation requests to determine whether it can reach an interface, or may listen in on acknowledgement packets of higher layer protocols, such as TCP.<br>If the node determines that a neighbor is no longer reachable, it attempts to reach the neighbor, or to re-establish the previous link. NUD can be used between any two devices in the network, independent of whether the devices are acting as hosts or routers. |

There are five IPv6 Neighbor Discovery messages that replace existing IPv4 messages:

Table 6: IPv6 replacement types

| IPV6 DISCOVERY MESSAGES | ICMPV6 TYPE | REPLACE THESE IPV4 MESSAGES |
|---|---|---|
| Router Solicitation<br>Router Advertisement | 133<br>134 | ICMPv4 Router Discovery |
| Neighbor Solicitation<br>Neighbor Advertisement | 135<br>136 | ARP |
| Redirect | 137 | ICMPv4 Redirect |

# Operation of neighbour and router discovery

## Neighbor solicitation

IPv6's replacement for ARP is Neighbor Solicitation, which uses two ICMP messages:

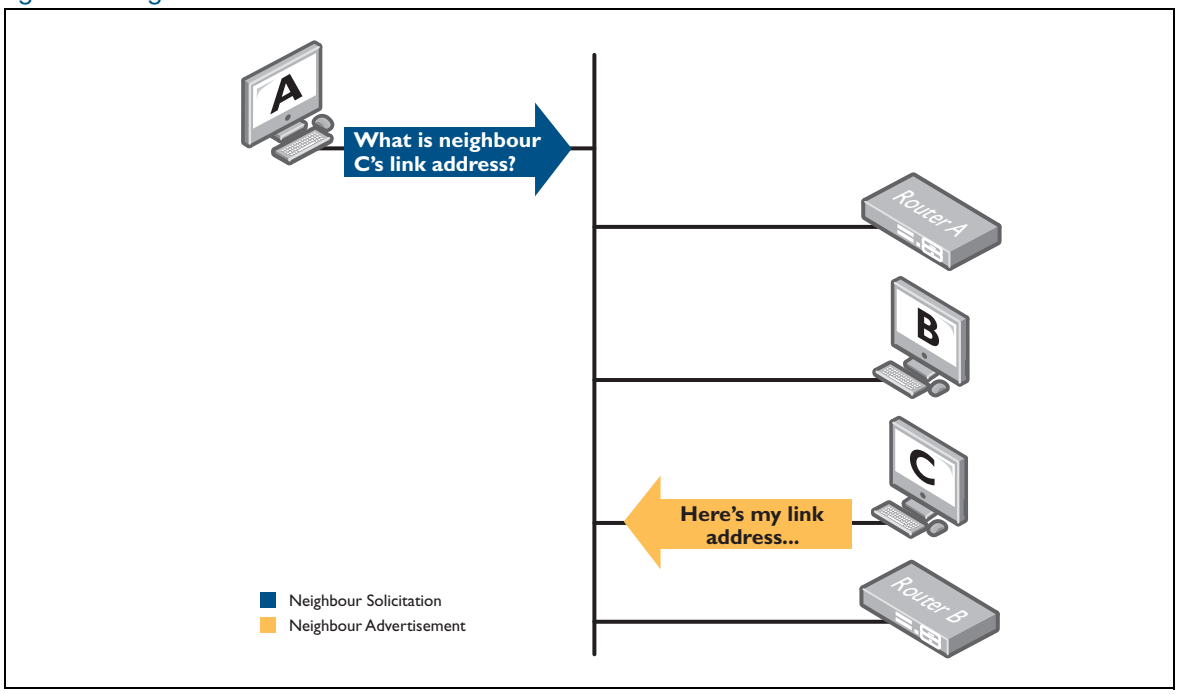
- Neighbor Solicitation (ICMPv6 Type 135)
- Neighbor Advertisement (ICMPv6 Type 136)

Neighbor solicitation messages perform the following functionality:

- They allow IPv6 nodes (IPv6 hosts and IPv6 routers) to resolve the Link Layer address of a neighboring node (a node on the same physical or logical link).
- When the Link Layer address of a neighboring node has changed, Neighbor Discovery messages allow the other IPv6 nodes to learn that this address has changed.
- They enable IPv6 nodes to determine whether neighboring nodes are still reachable.

In the diagram below, Host A sends a multicast packet (Neighbor Solicitation), and if Host C is operational it will respond to this packet with a Neighbor Advertisement packet.

Figure 2: Neighbor solicitation



## Solicited-node multicast address

When requesting the identity of the host that possesses a given IPv6 address, it is more efficient to multicast the request to potential candidates, rather than multicast to all hosts. This means that hosts that cannot possibly possess the address do not have to process unnecessary multicast packets. Solicited-node addresses are often flooded by switches and filtered by NIC cards drivers.

The multicast address used is called the solicited-node multicast address. It is created by attaching the last three bytes of the requested address to FF02::1:FF00:0

For example:

Address being requested: 2001: : 2AA:FF:F28:9C5A

1. Begin with FF02:0000:0000:0000:0000:0001:FF00:0000
2. Take the last 3 bytes of the requested address: 28:9C5A
3. Attached them to the address FF02::1:FF00:0

This is the solicited-node multicast address: FF02:0000:0000:0000:0000:0001:FF28:9C5A

## Router discovery

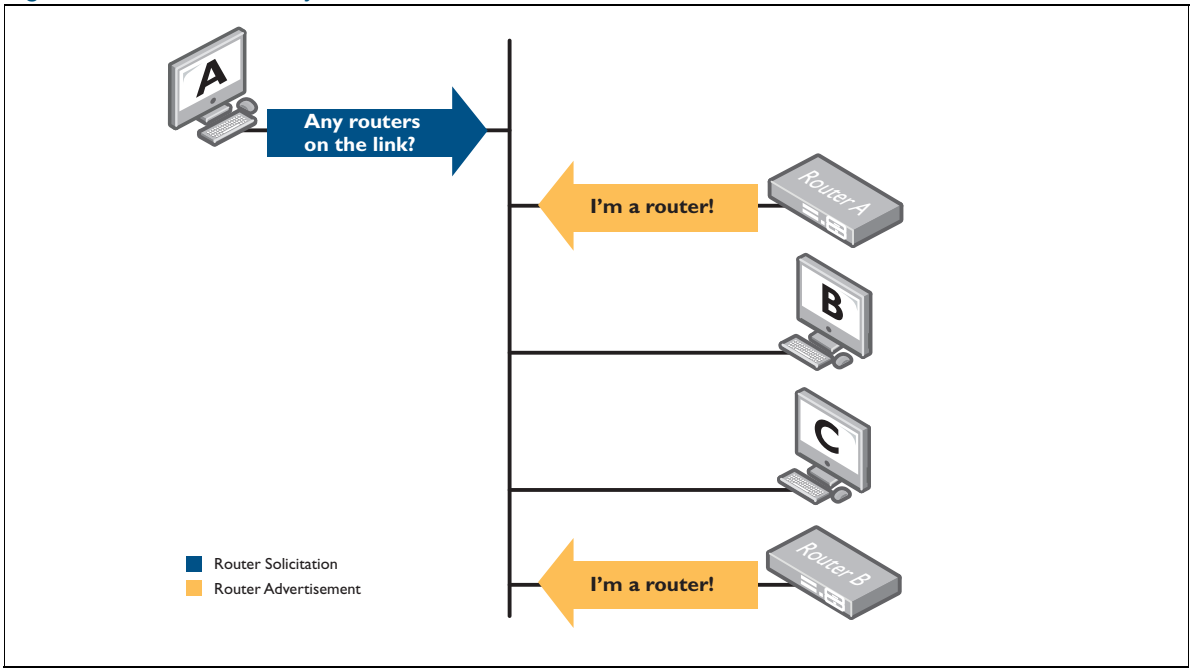
IPv4 hosts need either an administrator to manually configure the default gateway or DHCP to provide this information. When IPv6 is being used, the host themselves can automatically locate routers on the LAN. The host achieves this by using two different ICMPv6 messages.

They are:

- Router Solicitation (ICMPv6 Type 133)
- Router Advertisement (ICMPv6 Type 134)

When a host is first connected to a LAN, it will send an IPv6 Router Solicitation packet to request information about routers on the network. Each router which is active on the LAN will respond to this packet by sending a Router Advertisement (RA) with its address to all nodes in the group. It informs the host what network address(es) is (are) in use on the subnet. It also informs the host if it is a default gateway.

Figure 3: Router discovery




As well as responding to router solicitation events, a router will also send out RAs at regular intervals.

## Configuring router advertisements on AlliedWare Plus

RAs are configured on AlliedWare Plus on a per-interface basis.

To enable RAs use the command:

```
awplus(config-if)#no ipv6 nd suppress-ra
```

The options available are:

- IPv6 nd prefix<***x:x.../N***> which sets the prefix to advertise.
- IPv6 nd ra-interval <***seconds***> which sets the period of periodic RAs.
- IPv6 nd ra-lifetime <***seconds***> which sets the time for which the router will act as a default router, set this to zero to inform hosts that this is not a default router.

## Redirect

Redirect uses ICMP type 137 to inform a host of a better router to use as the gateway to a given destination. If a router receives a packet and has to forward that packet to another router in the same subnet, it will also send a redirect back to the sender, telling it to send directly to the other router.

# Stateless Address Autoconfiguration (SLAAC)

SLAAC allows an IPv6-aware device to be plugged into a network without manual configuration with an IP address. This plug and play functionality results in networks that are easier to set up and modify, and simplifies the process of shifting to use a new Internet Service Provider (ISP).

SLAAC is achieved using a series of steps. Routers and hosts perform the first three steps described below, which autoconfigure a link-local address. Then a global address is autoconfigured in the last three steps, which only hosts perform.

**Generate a link-local address on the router or host**

1. During system start-up, the device begins autoconfiguration by generating a **link-local address** for the interface. An EUI-64 link-local address is formed by adding the interface ID to the link-local prefix **fe80::/10** (reference RFC 3513). Alternatively, a link-local address can be statically configured.
2. The device performs Duplicate Address Detection (DAD) by transmitting a neighbor solicitation message to this address. If the address is already in use, the node that the address belongs to replies with a Neighbor Advertisement (NA) message. The autoconfiguration process stops and manual configuration of the node is then required.
3. If no neighbor advertisement is received, the node concludes that the address is available and assigns the link-local address to the chosen interface.

Note: The following steps assume the router is pre-configured with a global unicast IPv6 address, and with IPv6 Router Advertisement (RA) suppression, and Neighbor Discovery (ND) suppression **disabled** on the interface that the host is attached to.

**Configure a global address on the host**

4. The host then sends one or more router solicitations to detect if any routers are present. Any routers present respond with an RA. Routers may also periodically transmit RAs. If the router is also acting as a DHCPv6 server, it can optionally set the Managed (M) or Other Information (O) flags contained within the RAs, to inform any host devices that it can supply additional information, such as IPv6 address, or DNS addressing.

   If **no** RA is received, the host can attempt to use DHCP to obtain a globally scoped IPv6 address and other configuration information such as address or DNS. If no DHCPv6 server responds, the node continues using the link-local address only. For more information about DHCPv6, see the DHCPv6 Feature Overview Guide.

   If an RA is received, this message informs the host how to proceed with the autoconfiguration process. The prefix from the RA, if received, is concatenated with the link-level address to form the **global unicast IPv6 address**.
5. This address is then assigned to the network interface of the host. Additionally, the host dynamically creates a default route via the link-local address of the advertising router to provide connectivity via the gateway router to other networks.

   If routers are present, the host continues to receive periodic RAs. The host updates its configuration when there are changes in the RAs.

# Configuring SLAAC

There are two halves to the SLAAC configuration process—the client host side and the router side.

- **Client host side of SLAAC**—when an AlliedWare Plus device is performing as a client host, a VLAN interface has not been configured with an IPv6 address but instead learns an IPv6 address by SLAAC. To configure this mode on an interface, use the command:

```
awplus(config-if)#ipv6 enable
```

Note: Different interfaces on a device may have the same link-local address. The device will automatically generate a link-local address for all interfaces that are using IPv6. Commands entered to statically configure link-local addresses that match any automatically generated EU-64 link-local addresses by the device will not be executed. Enter the **show ipv6 interface** command to display automatically generated link-local addresses not shown in the running-config. Automatically generated link-local addresses contain the last six hexadecimal numbers of the MAC address for a given interface.

- **Router side of SLAAC**—when a client host is attached to the router, and the client hasn't been configured with an IPv6 address, the router can be configured to send out the network information in an RA (Router Advertisement) so the client is able to get an address and communicate on the LAN.

  For example, on a router's VLAN interface which has the client attached, the following configuration could be used to send the prefix:

```
awplus(config)#int vlan10
awplus(config-if)#ipv6 address 2001:db8:2::1/64
awplus(config-if)#ipv6 nd ra-interval 10
awplus(config-if)#ipv6 nd prefix 2001:db8:2::/64
awplus(config-if)#no ipv6 nd suppress-ra
```

Note: AlliedWare Plus supports both the client host and the router side of SLAAC.

Note: From software release 5.4.7-1.x onwards, the command **ipv6 address autoconfig** (used to enable SLAAC on an interface) has been deprecated, and is no longer necessary to configure. SLAAC is now implicitly **enabled** when any one of the following commands is configured on an interface:

  - ipv6 enable
  - ipv6 address
  - ipv6 address dhcp

## Important security considerations when enabling IPv6

By default when you enable IPv6 on an interface by any one of the following commands, SLAAC is implicitly enabled:

- ipv6 enable
- ipv6 address <***ipv6-addr/prefix-length***>
- ipv6 address dhcp

This means that any Router Advertisement (RA) received on that interface will be processed, potentially resulting in the assignment of IPv6 addresses and learning default routes. In most situations this is the desired behavior. However, in some circumstances, particularly if using static address assignment, this may not be desirable or may present a security risk.

The security risk is that a person with access to the connected network could inject an RA into the network. This can trigger the device to add an IPv6 address or a default route which could be used for malicious purposes.

To prevent this from happening in an environment where RA processing is not required on an interface, we recommend disabling RA processing, using both the following commands:

- no ipv6 nd accept-ra-default-routes
- no ipv6 nd accept-ra-pinfo

**Example**

To configure a static IPv6 address on vlan1 and then disable processing of RA's to prevent malicious activity, use the following steps:

Step 1: **Configure a static address on vlan1**

```
awplus#configure terminal
awplus(config)#int vlan1
awplus(config-if)#ipv6 address 2001:db8:1::1/64
```

Step 2: **Disable RA processing**

```
awplus(config-if)#no ipv6 nd accept-ra-default-routes
awplus(config-if)#no ipv6 nd accept-ra-pinfo
```

## Setting up an IPv6 interface using the EUI-64 algorithm

Here is an overview of the steps that occur when a device performs SLAAC using the EUI- 64 algorithm:

**1.** Generate a 64 bit interface identifier using the EUI-64 algorithm.

The host has to create its own host portion of its IPv6 address. This is commonly known as the EUI-64 link-local address. It can create a unique address from its MAC address by using the EUI-64 algorithm, here is how it works:

Table 7: Generate a 64 bit interface identifier

| STEP | ADDRESS |
|---|---|
| 1. Start with the MAC address | 0012.7FEB.6B40 |
| 2. Split the MAC address in half | 0012:7F EB:6B40 |
| 3. Insert FF:EE into the MAC address | 0012:7FF**F:FEE**B:6B40 |
| 4. Change the 7th bit from the left to '1' | 0212:7FFF:FEEB:6B40 |

**2.** Assign a link-local node address.

When IPv6 has been configured on an interface, the device will automatically assign a link-local address to that interface. Link-local addresses are used as the source address for packets that stay within the subnet, for example:

- automatic address configuration
- neighbor discovery
- OSPF exchanges etc.

Any packets that are transmitted with a link-local source/destination address are never routed out of that subnet and are assigned the fe80::/10 prefix, equivalent to the IPv4 address block 169.254.0.0/16.

The link-local address for an interface is created by combining the EUI-64 host address to the network address FE80::/64.

FE80:0000:0000:0000: + 0212: 7FFF:FEEB:6B40= FE80: :0212: 7FFF:FEEB:6B40

**3.** Send router solicitation messages to all routers on the local link multicast address. If there is no response, SLAAC ends with only a link-local address generated.

Note: If a periodic RA containing the appropriate information is received at any time, SLAAC will use it immediately. Also, if an RA is received that reduces the lifetime of a prefix to zero, SLAAC will immediately deprecate the address (the system will then cease using it for new connections, but existing ones will continue).

4. Once a prefix is learnt by RA, prepend the prefix to the EUI-64 interface ID, to create the full global unicast IPv6 address.

   2001:639A:1234:5678:: + 0212: 7FFF:FEEB:6B40

   =

   2001:639A:1234:5678:0212:7FFF:FEEB:6B40

5. Find default gateway (default routers). On receipt of a valid RA, a host extracts the source address of the packet and does the following:

- If the address is not already present in the host's Default Router List, and the advertisement's Router Lifetime is non-zero, it creates a new entry in the list and initializes its invalidation timer value from the advertisement's Router Lifetime field.
- If the address is already present in the host's Default Router List as a result of a previously received advertisement, it resets its invalidation timer to the Router Lifetime value in the newly received advertisement.
- If the address is already present in the host's Default Router List and the received Router Lifetime value is zero, it immediately times-out the entry as specified.

To limit the storage needed for the Default Router List, a host may choose not to store all of the router addresses discovered via advertisements. However, a host must retain at least two router addresses and should retain more. Default router selections are made whenever communication to a destination appears to be failing. Thus, the more routers on the list, the more likely an alternative working router can be found quickly (without having to wait for the next advertisement to arrive).

## Encryption and authentication in IPv6

IPv4 protocols such as OSPFv2, have authentication incorporated into their own protocol header.

In IPv6, authentication and encryption are performed by separate IP headers, completely independent to the enclosed protocol.

## AH – Authentication Header – commonly MD5 or SHA

The authentication information for the Authentication Header is calculated using all the fields of the datagram that do not change in transit.

This header can be used as part of IPSec to authenticate end point to end point packets. This can be used to protect protocols like OSPFv3, IPv6, BGP, RADIUS, TACACS+, and RIPng.

## ESP – Encapsulated Security Payload – commonly 3DES or AES

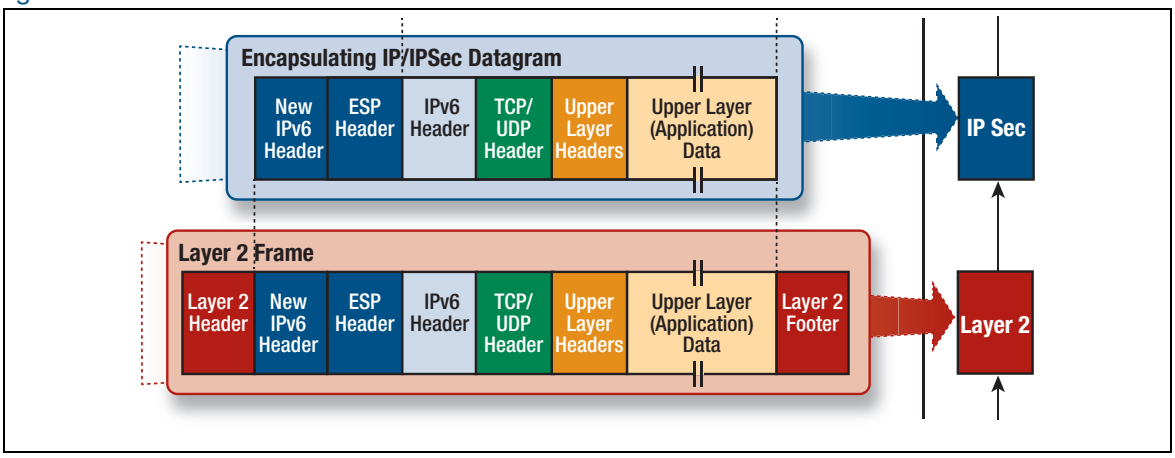
ESP is used to convey the encrypted data of the IP datagram. The encrypted data is obtained by applying a specified encryption transform to the data and requires the use of a key in order to return to plain text.

There are two modes used for ESP:

- Tunnel Mode
- Transport mode

**Tunnel Mode**, where the **entire IP packet** is encrypted and/or authenticated. It is then encapsulated into a new IPv6 packet with a new IPv6 header. ESP Tunnel mode encrypts the whole IPv6 datagram:

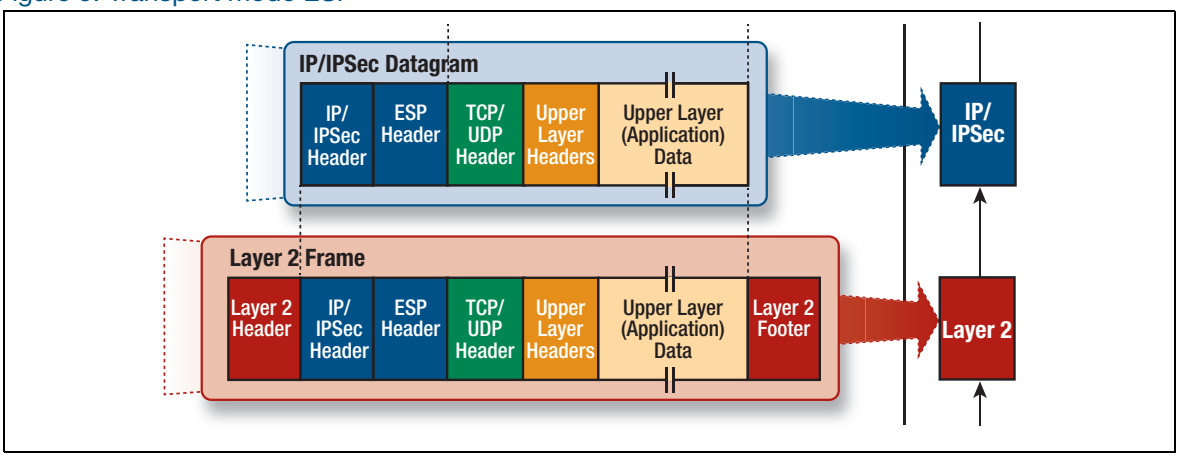
Figure 4: Tunnel mode ESP




In ESP Tunnel mode, the Authentication Header appears as an extension header of the new IPv6 datagram that encapsulates the original one being tunneled.

**Transport Mode**, where **only the payload** of the IPv6 packet is encrypted and/or authenticated – not the original IPv6 Header. ESP Transport mode encrypts only the payload (Transport Layer message of the IPv6 datagram):

Figure 5: Transport mode ESP



The extension headers used to secure the IPv6 communication between two hosts, Encapsulating Security Payload Header, is ignored by the intermediary network devices while forwarding traffic. This Extension Header is relevant only to the source and destination of the IPv6 packet.

All information following the ESP Header is encrypted and not available for inspection by an intermediary device.

## The Flow label

The Flow label is a 20 bit field in the IPv6 packet header which provides an efficient way for packet marking, flow identification, and flow state lookup.

This field can be used by a source to label a set of packets belonging to the same flow. The device must process the packets in the same flow in the same manner. When a flow-label aware router receives the first packet of a new flow, it sets up a new flow entry using the information carried by the IPv6 header, Routing header, and Hop-by-Hop extension headers, and stores the result.

It then uses the flow entry to route all other packets belonging to the same flow – which will have the same source address and the same Flow Label.

## IPv6 routing

Routing in IPv6 is almost identical to IPv4 routing under CIDR, except that the addresses are **128-bit** IPv6 addresses instead of 32-bit IPv4 addresses.

**Routing Information Protocol (RIPv6)**

RIP is a simple distance vector protocol that defines networks based on how many hops they are from the router. When a network is more than 15 hops away (one hop is one link), it is not included in the routing table.

RIPv6, also referred to as RIPng (for “next generation”) is similar to RIPv2. Extensions to RIPv2 to support IPv6 are:

- the address field of a routing entry is expanded to 128 bits to allow IPv6 prefixes
- the 32-bit RIPv2 subnet mask field is replaced by an 8-bit prefix length field
- authentication is removed in RIPv6
- the size of a routing packet is no longer arbitrarily limited
- RIPv6 specifies the next hop instead of simply allowing the recipient of the update to set the next hop to the sender of the update.

In RIPv6, each router uses a routing table to keep track of every destination that is reachable throughout the system. Each entry in the routing table contains:

- the IPv6 prefix of the destination
- a metric, which represents the total cost of getting a packet from the router to that destination
- the IPv6 address of the next router along the path to the destination
- a flag to indicate that information about the route has changed recently
- various timers associated with the route.

## Configuring IPv6 routing

IPv6 Layer 3 forwarding is disabled by default. To enable IPv6 forwarding, use the **ipv6 forwarding** command.

To display information about IPv6 settings, use the **show ipv6 interface brief** command. Because AlliedWare Plus implements IPv6 as a dual protocol stack, implementing IPv6 does not affect IPv4 functionality.

## IPv6 interface states

From AlliedWare Plus version 5.4.8 onwards, the **show ipv6 interface** and **show interface** commands also return information about the configuration and state of connected IPv6 addresses.

**What are the IPv6 address states?**

The IPv6 address states are: tentative, DAD failed, preferred, and deprecated.

- **Tentative** - an address in the process of being verified through duplicate address detection.
- **DAD failed** - duplicate address detection found that the address is not unique and cannot be used on this interface.
- **Preferred** - an address that has been verified as unique. Communication with this address is unrestricted.
- **Deprecated** - if the preferred lifetime of a preferred address times out the address goes into the deprecated state. Communication to/from a deprecated address is valid but discouraged.

**You can see interface state information in the show command output**

The command **show ipv6 interface** output includes the:

- **State** field heading
- State of each address - the star symbol ‘*’ indicates an auto-configured address

```
awplus#show ipv6 interface
  * = Autoconfigured Address
Interface    IPv6-Address                 State       Status     Protocol
lo           unassigned                   N/A         admin up   running
vlan1        2001:4::124:3/64             tentative   admin up   running
               2001:dead:beef:0:230:abff:fef3:42b4/64  deprecated*
               fe80::222:b0ff:fede:803c/64     DAD failed
               2001:dead:beef:0:21a:ebff:fe94:27a7/64  preferred *
               fe80::21a:ebff:fe94:27a7/64     preferred
```

The command **show interface** output includes the:

- State of each IPv6 address - the star symbol '*' indicates an auto-configured address.

```
awplus#show interface vlan1
Interface vlan1
  Scope: both
  Link is UP, administrative state is UP
  Hardware is VLAN, address is 0030.abf3.42b4
  IPv6 address 2001:4::124:3/64 tentative
  IPv6 address 2001:dead:beef:0:230:abff:fef3:42b4/64 deprecated*
  IPv6 address fe80::222:b0ff:fede:803c/64 DAD failed
  IPv6 address 2001:dead:beef:0:21a:ebff:fe94:27a7/64 *
  IPv6 address fe80::21a:ebff:fe94:27a7/64
  index 301 metric 1 mtu 1500
  arp ageing timeout 300
  <UP,BROADCAST,RUNNING,MULTICAST>
  SNMP link-status traps: Disabled
  Router Advertisement is disabled
  Router Advertisement default routes are accepted
  Router Advertisement prefix info is accepted
    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 00:40:07
```

**Interface state information is useful for network engineers**

State information is useful because only **preferred** or **deprecated** addresses are valid for sending and receiving.

IPv6 addresses can be configured manually or automatically. Auto-configured address are temporary and should not be treated in the same way as permanent addresses because they may change. So, it is useful to be able to easily tell if an address is auto-configured as well as the IPv6 state of the address.

Expanding the output of the **show ipv6 interface** and **show interface** commands to include IPv6 states and distinguishing between permanent and non-permanent addresses, enables network engineers to learn information about administrative, link, and ipv6 state, all in one place.

## Integration of IPv4 and IPv6

IPv6 has been designed in such a way that a smooth transition from IPv4 is possible. The most effective way to ensure this is to use a ***dual IP stack***. A node configured as a dual stack system has both a 128-bit IPv6 address and a 32-bit IPv4 address, and so can communicate with nodes running IPv4 and those running IPv6.

# IPv6 RA Guard

Router Advertisements (RA) and Router Redirects are key to the Network Discovery Protocol (NDP) which is used to manage IPv6 networks. RA messages advertise a router's presence and specify network parameters that are used by hosts as part of address auto-configuration and next hop routers for particular destinations.

Subverting this process can severely disrupt the operation of an IPv6 network. RA Guard is a feature that protects the RA process from being subverted.

RA Guard:

- is positioned in between routers and hosts, and acts as an authorisation proxy.
- operates on all AlliedWare Plus Layer 3 switches, including stacked environments.
- drops bad or Rogue RAs before they reach hosts.

## Rogue RAs

A rogue RA is an RA that contains invalid information that could cause unwanted changes in the network configuration. These could be generated unintentionally through misconfiguration or maliciously by someone wanting to disrupt or gain access to the network.

A switch can be configured to be selective about the RA and redirect packets it will accept. Ports are configured to trust or not trust the RA and redirect packets they receive.

## RA Guard on AlliedWare Plus switches

Ports can be configured to be RA untrusted ports, i.e. RA Guard is applied to ports on a per-interface basis and can be enabled on the following:

- Standalone ports.
- Individual ports in a dynamic (LACP) aggregator, but is not supported on the dynamic aggregator itself.
- A static aggregator, but is not supported on individual ports in a static aggregator.

RA Guard is enabled on an interface as follows:

```
awplus#conf terminal
awplus(config)#interface port1.0.2
awplus(config-if)#ipv6 nd raguard
```

Note: This feature is disabled by default on IPv6 enabled interfaces.

## RA Guard classifiers

The actual security enforcement of RA Guard is handled through hardware classifiers, which are dynamically added when a port is marked as trusted or untrusted.

RA Guard blocks RAs and router redirects on untrusted ports with filters for ICMPv6 type 134 and 137.

## Enabling IPv6 RA Guard

IPv6 RA Guard is disabled by default. To enable IPv6 RA Guard on a port to block RAs from an untrusted host, use the **ipv6 nd raguard** command. Disable IPv6 RA Guard to allow RAs on a port using the **no ipv6 nd raguard** command.

# IPv6 Source Address Dependent Routing (AR-Series only)

## Introduction

IPv6 Source Address Dependent Routing (SADR) can be useful where there are two or more active IPv6 WAN links originating from an AlliedWare Plus Firewall router, with each WAN link connecting via different ISPs.

Each upstream ISP router typically only accepts traffic originating from global scoped prefixes they have allocated for use. All other traffic originating from other prefixes associated with a different ISP is denied.  IPv6 SADR allows routes to be used for a subset of all prefixes. This ensures traffic sourced from prefixes allocated by an ISP is routed only to that ISP.

## How does IPv6 SADR work?

IPv6 routers advertise their capabilities via Router Advertisement (RA) packets. These RA packets indicate that the originating device may be used as a default route, and may provide prefix information indicating which IP addresses may be used by devices on that link.

When searching through the forwarding rules for a nexthop to forward a packet to, selection is usually based strictly on the destination address of the packet. IPv6 SADR allows inspection of the source address and ensures it fits within the valid source prefix provided for the route. This is important when a device has two or more upstream routers.

IPv6 SADR routes are configured either statically or dynamically, in response to PrefixInfo options provided by the upstream device in an RA.

## Static routes

Static IPv6 SADR routes allow traffic to be separated manually and sent to the correct nexthop.

```
ipv6 ::/0 2001:db8:1::1 eth1 2001:DB8:1::/64
ipv6 ::/0 2001:db8:2::1 eth2 2001:DB8:2::/64
```

In this example above:

- traffic sourced from 2001:DB8:1::/64 would use the default route to 2001:DB:1: : 1 on eth1
- traffic sourced from 2002:DB8:1::/64 would use the default route to 2001:DB:2: : 1 on eth2
- traffic sourced from other subnets would not have a default route available

## Dynamic routes

Use the **ipv6 multihoming** command to allow a device to add a dynamic SADR default route when it receives a Router Advertisement with Prefix Information options provided. These Prefix Information options indicate which global IPv6 prefixes may be assigned to devices on the link. These are the prefixes that the upstream router can expect to receive traffic from.

## Configuring SADR

Use the **ipv6 multihoming** command to configure dynamic routes. Dynamic routes will almost always be the correct way to configure the SADR feature, as in this example:

```
interface eth1,eth2
  ipv6 enable
  no ipv6 nd accept-ra-default-routes
  ipv6 multihoming
```

Use the **no ipv6 multihoming** to remove any dynamic routes learned on that interface.

## Monitoring SADR

Use the **show IPv6 route** command to display the SADR information (if present).

```
awplus#show ipv6 route
Codes: C - connected, S - static, R - RIP, B - BGP
       O - OSPF, D - DHCP, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2

S       ::/0 from 2001:db8:1::/64 [1/0] via 2001:db8:1::1, eth1
S       ::/0 from 2001:db8:2::/64 [1/0] via 2001:db8:2::1, eth2
C       2001:db8:1::/64 is directly connected, eth1
C       2001:db8:2::/64 is directly connected, eth2
```

# IPv6 Neighbor Discovery Proxy (AR-Series only)

IPv6 Neighbor Discovery (ND) Proxy is often used as a method to provide mobile IPv6 connectivity via 3GPP networks. This technology also provides IPv6 over Ethernet (IPv6oE) Internet connectivity to some service providers.This feature implementation is limited to the components of Neighbor Discovery Proxy RFC 4389. Therefore, not all capabilities are described within the RFC are supported in AlliedWare Plus.

The IPv6 ND proxy feature is supported in AR-Series AR2010V, 2050V, AR3050S and AR4050S models from version 5.4.6-2 onwards. This feature allows for IPv6 addressing information to be dynamically learned via an external Ethernet WAN interface. For example, via SLAAC or alternatively via DHCPv6 from an ISP router. This address information is assigned to an internal LAN interface, instead of being used on the WAN.

## WAN assignment via SLAAC with IPv6 ND proxy

In order for this to work, IPv6 NS/NA messages need to be proxied and forwarded (bridged) between a specific pair of WAN and VLAN interfaces.

To proxy these messages, use the interface configuration command:
**ipv6 nd proxy interface <*interface-name*>**.

This command enables the ND proxy that forwards Neighbor Solicitations (NS) and Neighbor Advertisements (NA) between the two interfaces that have been configured.

More specifically, this command enables the forwarding of NS between the related internal (VLAN) to external (Ethernet WAN) interfaces. Then, related forwarding and matching of NA that comes back in reply. This is required to support IPv6 Duplicate Address Detection (DAD).

You need to configure the command **ipv6 address autoconfig eth1** on the VLAN interface to allow the VLAN IPv6 address to be acquired via SLAAC, and assigned based on RA received via the Ethernet WAN.

Additionally, a default route is automatically and dynamically created, with the nexthop IPv6 address based on the source link-local address of the packets from the device advertising the RAs.

Optionally, if the service provider advertises RAs with O Flag “set”, this prompts the AR-Series Firewall to automatically request and learn DNS server information via the WAN.

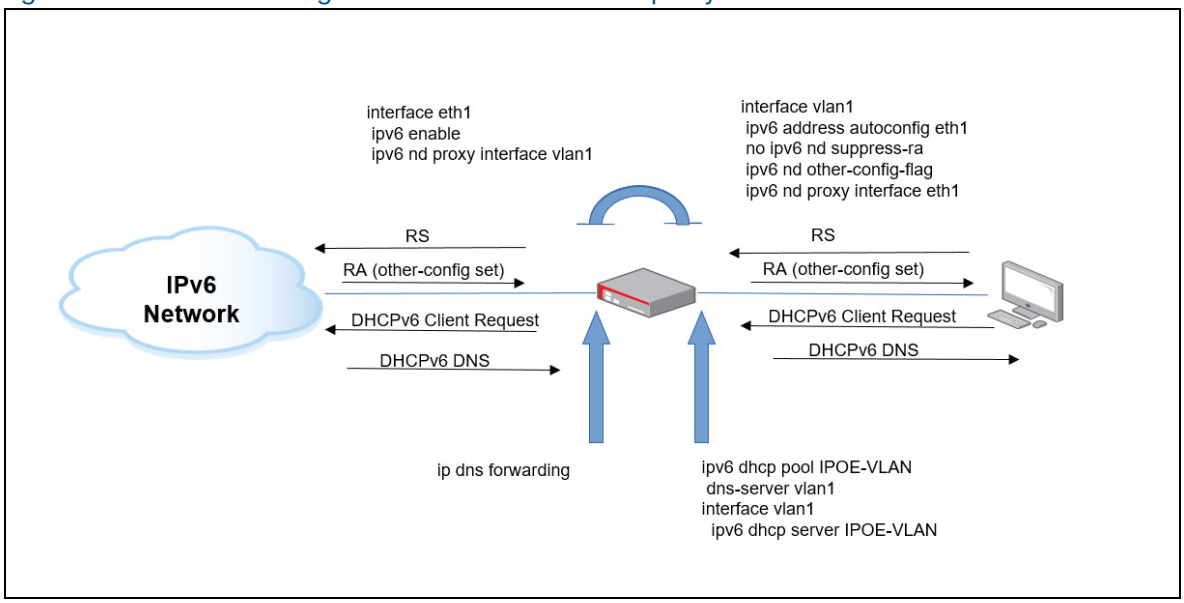
If DHCv6 server components on the LAN are configured, (such as **ipv6 dhcp server**, **no ipv6 nd suppress-ra** and, **ipv6 nd other-config-flag**) then the router sends RAs to the LAN interface with the “other config” flag set. This prompts the LAN hosts to request other configuration via DHCPv6.

The router advertises itself to the LAN as the DNS server, via DHCPv6. The presence of the “other config” flag within an RA, indicates to the host that there is additional configuration available (DNS, SNTP) via a stateless DHCPv6 client request. Client DNS requests are sent to the LAN address of the AR-Series Firewall.

You need to configure the DNS forwarding cache, in oder for the AR-Series Firewall to perform DNS lookups on behalf of client requests, and store the results.

The following diagram shows key components of the feature set “IPv6oE WAN assignment via SLAAC with ND proxy”:

Figure 6: IPv6oE WAN assignment via SLAAC with ND proxy




The following configuration shows how to configure WAN assignment via SLAAC with ND proxy:

```
!
ipv6 dhcp pool IPoE-vlan1
 dns-server interface vlan1
!
interface eth1
 ipv6 enable
 no ipv6 nd accept-ra-pinfo
 ipv6 nd proxy interface vlan1
!
interface vlan1
 ipv6 address autoconfig eth1
 no ipv6 nd suppress-ra
 ipv6 nd other-config-flag
 ipv6 nd proxy interface eth1
 ipv6 dhcp server IPoE-vlan1
!
ipv6 forwarding
!
ip dns forwarding
ip dns forwarding source-interface vlan1
!
```

## WAN assignment via DHCP prefix delegation - without IPv6 ND proxy

The following shows how to configure DHCPv6 prefix delegation as an alternative for WAN address assignment, instead of using SLAAC plus IPv6 ND proxy. In this example, ND proxy is not required.

The Ethernet WAN is optionally configured as a DHCPv6 prefix delegation client interface, and it is also configured to create an IPv6 default route dynamically, based on the IPv6 source IP address DHCPv6 address of the DHCPv6 server.

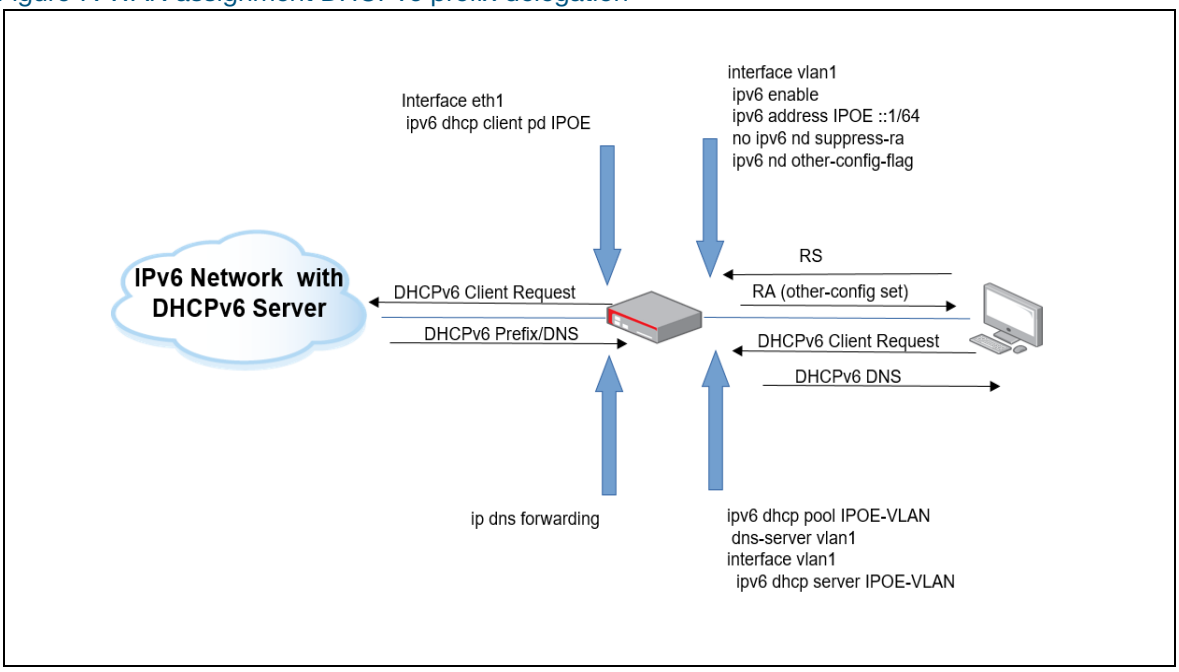
- Since the default route is configured based on DHCPv6 server messages, the Ethernet WAN is configured with the non-default option to ensure the default IPv6 route is not installed into the routing table based on RAs received via the Ethernet WAN. Only the link-local address is assigned to the Ethernet WAN.
- The prefix advertised from the DHCPv6 server is stored within the configured DHCPv6 address pool, this prefix information is used to build the globally scoped IPv6 address allocated to the VLAN.
- Additionally, the DHCPv6 server is optionally configured to provide stateful IPv6 addressing to clients located on the VLAN. This is once the VLAN RA suppression is turned off, and RAs are sent with the O flag “set”, to inform clients on the LAN of the availability of stateful address assignment via DHCPv6.
- Optionally, the command **no ipv6 nd prefix auto-advertise** can be configured. This command is used in conjunction with the command **no ipv6 nd supress-ra** to disable the advertisement of link prefixes. A specific prefix can then instead be selectively advertised. For example, via DHCPv6, if required. The default behavior is to automatically advertise on link address prefixes.

The following configuration shows how to configure DHCPv6 prefix delegation client

```
!
ipv6 dhcp pool IPOE-VLAN
 dns server vlan1
!
interface eth1
 ipv6 enable
 ipv6 dhcp client pd WANPREFIX default-route-to-server
 no ipv6 nd accept-ra-default-routes
!
interface vlan1
 ipv6 address WANPREFIX ::bbbb/64
 no ipv6 nd suppress-ra
 no ipv6 nd prefix auto-advertise
 ipv6 nd other-config-flag
 ipv6 dhcp server IPOE-VLAN
!
```

The following diagram shows key components of the feature set DHCPv6 server configuration used to advertise the DHCPv6 prefix towards the DHCPv6 client:

Figure 7: WAN assignment DHCPv6 prefix delegation




The DHCPv6 server configuration is as follows:

```
!
hostname DHCPv6SERVER
!
ipv6 local pool test 2001:db8::/60 64
!
ipv6 dhcp pool poolname
 prefix-delegation pool test
!
interface eth1
 ipv6 address 2001:db8::1/64
 ipv6 dhcp server poolname
!
ipv6 forwarding
!
```

For more information about DHCPv6, see the DHCP for IPv6 Feature Overview and Configuration Guide.

## WAN assignment via SLAAC with IPv6 ND proxy, DNS and firewall

The following example shows how to configure WAN assignment via SLAAC with IPv6 ND proxy between an internal VLAN and an external Ethernet WAN interface.

Firstly, in this example, the IPv6 firewall is used optionally, so configure the firewall internal Internet zone entities and applications to match DHCPv6 and ICMPv6 traffic. You can then configure firewall rules to permit all IPv6 traffic within the internal zone. Also, you can permit all IPv6 traffic to reach the Internet, and to allow ICMPv6 and DHCPv6 via the external Ethernet WAN.

Rules can be checked via the **show firewall rule** and **show firewall rule config-check** commands. DHCPv6 rules are necessary because IPv6 DNS information is learned via stateless DHCPv6 via the external Ethernet WAN from the service provider.

In this example, global scoped IPv6 addressing is accomplished based on RAs and SLAAC, not DHCPv6.

This example has a service provider router that is also configured as a DHCPv6 server. It is located within the service provider network.

- The provider's DHCPv6 server pool is configured to advertise IPv6 DNS server addresses only.
- The DHCPv6 server is not configured to advertise IPv6 addressing/prefix information.
- The service provider router is also configured to advertise IPv6 Router Advertisements (RAs) with the O flag "set" and sent out its interface towards the AR-Series Firewall. The M flag is "not set" because the DHCPv6 server is not performing stateful IPv6 address/prefix delegation. The setting of the O flag in the RAs informs devices that they can perform stateless DHCPv6, to obtain "other" information, such as DNS server, or SNTP.

On receipt of the RAs from the service provider router, stateless DHCPv6 is used automatically by the AR-Series Firewall to request and learn the IPv6 address of the DNS server (via the Ethernet WAN), from the service provider DHCPv6 server. Use the command **show ip name-server** to view the dynamically learned DNS server address.

Next, you need to configure an IPv6 DHCP pool. The pool is configured with the VLAN interface because that is the interface which has the globally scoped address. Additionally, you can then configure the IPv6 DHCPv6 server on the internal VLAN of the AR-Series Firewall.

Now you can additionally disable RA suppression and ensure that the O flag only is "set" (not the M flag), on the internal VLAN interface of the AR-Series Firewall. This ensures that the AR-Series Firewall DHCPv6 server advertises the globally scoped VLAN interface address as the DNS server, on receipt of the stateless DHCPv6 request from any client computers.

Next, you can configure the internal VLAN to use SLAAC, with the globally scoped IPv6 address assigned to the internal VLAN interface. This is based on RAs received via eth1 WAN from the service provider router. eth1 WAN is automatically configured with link-local only.

Use the command **show ipv6 interface**, and **show interface brief** commands to view the IPv6 addressing, and interface information. The AR-Series Firewall automatically creates an IPv6 default route via the next-hop link-local address of the service provider router. This happens upon receipt of the RAs via the Ethernet WAN. This is visible via the **show ipv6 route** command.

With the AR-Series Firewall now itself advertising RAs with the O flag “set”, the VLAN and any host computers attached to the internal VLAN interface of the AR-Series Firewall can themselves automatically use SLAAC to configure their own globally scoped address.

Attached host computers will also automatically create an IPv6 default route to the link-local gateway address of the AR-Series Firewall VLAN interface. This is based on the RAs sent by the AR-Series Firewall.

Because the O flag is “set” in the RAs from the AR-Series Firewall, attached host computers also automatically perform stateless DHCPv6 and automatically learn that the AR-Series Firewall is their IPv6 DNS server.

IPv6 ND proxy is configured to ensure IPv6 ND/NS messages are proxied between VLAN and Ethernet interfaces. This allows any attached host computers to perform Neighbor Discovery/ Solicitation and also performs checks for any Duplicate Address Detection (DAD), via the WAN and LAN.

Lastly, configure IPv4/IPv6 DNS forwarding cache. This ensures the AR-Series Firewall performs DNS relay/lookup on behalf of DNS requests from host computers and locally caches the results.

Use the command **show ip dns forwarding cache** to view the DNS cache entries.

Host computers attached to the LAN have a DNS entry pointing to the globally scoped IPv6 address of the AR-Series Firewall LAN, obtained via stateless DHCPv6.

Host computers point to the AR-Series Firewall VLAN link-local address as their IPv6 gateway, obtained via RAs from the AR-Series Firewall. Any traffic to the Internet or IPv6 DNS lookups are directed via the AR-Series Firewall.

The following configuration shows how to configure WAN assignment via SLAAC with ND proxy, DNS and Firewall:

```
!
zone ipv6-internal
 network lan
  ipv6 subnet ::/0 interface vlan1
  host vlan1
   ipv6 address dynamic interface vlan1
!
zone ipv6-internet
 network wan
  ipv6 subnet ::/0 interface eth1
  host eth1
   ipv6 address dynamic interface eth1
!
application dhcpv6-r
 protocol udp
 sport 547
 dport 546
!
application dhcpv6-s
 protocol udp
 sport 546
 dport 547
!
application icmpv6
 protocol ipv6-icmp
!
firewall
 rule 10 permit any from ipv6-internal to ipv6-internal
 rule 20 permit any from ipv6-internal to ipv6-internet
 rule 30 permit dhcpv6-s from ipv6-internet.wan.eth1 to ipv6-internet
 rule 40 permit dhcpv6-r from ipv6-internet to ipv6-internet.wan.eth1
 rule 50 permit icmpv6 from ipv6-internet to ipv6-internal.lan
 protect
!
ipv6 dhcp pool ipv6
 dns-server interface vlan1
!
interface eth1
 ipv6 enable
 no ipv6 nd accept-ra-pinfo
 ipv6 nd proxy interface vlan1
!
interface vlan1
 ipv6 enable
 ipv6 address autoconfig eth1
 no ipv6 nd suppress-ra
 ipv6 nd other-config-flag
 ipv6 nd proxy interface eth1
 ipv6 dhcp server ipv6
!
ipv6 forwarding
!
ip dns forwarding
ip dns forwarding cache size 1000 timeout 3600
!
```