Introduction

This guide describes how to configure IPv4 addressing and the protocols used to help IP function on your network.

As well as the familiar Internet (with uppercase “I”), the term internet (with lowercase “i”) can refer to any network (usually a wide area network) that uses the Internet Protocol. This guide concentrates on this definition—a generalized network that uses IP as its network protocol.

Products and software version that apply to this guide

This guide applies to all AlliedWare Plus™ products, running version 5.4.4 or later.

However, feature support and implementation varies between products. To see whether a product supports a particular feature or command, see the following documents:

- The product’s Datasheet
- The AlliedWare Plus 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

Introduction ........................................................................................................................ 1  
  Products and software version that apply to this guide ............................................. 1  
Assigning an IP Address ................................................................................................. 3

**Address Resolution Protocol (ARP)**........................................................................ 4  
  Static ARP entries ....................................................................................................... 4  
  Timing out ARP entries ............................................................................................... 4  
  Deleting ARP entries .................................................................................................. 5  
  Proxy ARP..................................................................................................................... 5  
  ARP logging................................................................................................................... 8

**Domain Name System (DNS)** .................................................................................... 8  
  Domain name parts ..................................................................................................... 9  
  Server hierarchy ......................................................................................................... 9  
  DNS client..................................................................................................................... 10  
  DNS Relay................................................................................................................... 11  
  DNS domain name matching ..................................................................................... 13  
  DNS operation with VRF-lite ..................................................................................... 15  
  DHCP options............................................................................................................. 16  
  DHCP operation with VRF-lite ................................................................................... 16

**Internet Control Message Protocol (ICMP)** ............................................................ 17  
  ICMP Router Discovery Protocol (IRDP) .................................................................... 18  
  Router Discovery ....................................................................................................... 18  
  Router Discovery process ......................................................................................... 18  
  Configuration procedure............................................................................................. 20

**Checking IP Connections** ........................................................................................ 22  
  Ping............................................................................................................................. 22  
  Traceroute ................................................................................................................... 22

**IP Helper (UDP Broadcast Helper)** .......................................................................... 23

**IP Directed Broadcast** .............................................................................................. 24

**Support for Network Load Balancing (NLB)** .......................................................... 24  
  Introduction................................................................................................................ 24  
  Network Load Balancing clustering modes ................................................................ 24  
  Multicast mode.......................................................................................................... 28  
  Multicast mode with the IGMP option selected ....................................................... 31  
  Support for NLB in x-Series Switches ....................................................................... 34
Assigning an IP Address

To configure your device to perform IP routing (for example, to access the Internet) you need to configure IP. You also need to configure IP if you want to manage your device from any IP-based management process (such as SSH, Telnet, or SNMP).

Add an IP address to each of the interfaces to or from which you want to route IP, or which you want to use as a management interface on the device. You can configure an interface on your device with a static IP address, or with a dynamic IP address assigned using your device’s DHCP client.

**Static IP addresses**

To add a static IP address to an interface, enter interface mode for the interface that you want to configure, then use the command:

```
awplus(config-if)# ip address <ip-addr/prefix-length> [secondary] [label <label>]
```

where `<ip-addr/prefix-length>` is the IP address followed by a slash then the prefix length. Note that you cannot specify the mask in dotted decimal notation in this command.

**Note:** On SBx8100 Series switches, the subnet 192.168.255.0/28 is internally reserved and cannot be configured.

For example, to give the interface vlan1 an address of 192.168.10.10, with a class C subnet mask, use the command:

```
awplus(config-if)#ip address 192.168.10.10/24
```

The `secondary` parameter allows you to add multiple IP addresses to an interface using this command. Each interface must have a primary IP address before you can add a secondary address. Your device treats secondary addresses the same as primary addresses in most respects, such as responding for ARP requests for the IP address. However, the only packets generated that have a secondary address as source address are routing updates. You can define up to 32 secondary addresses on a single interface.

**DHCP dynamic addresses**

When you use the DHCP client, it obtains the IP address and subnet mask for the interface, and other IP configuration parameters, from a DHCP server. To configure an interface to gain its IP configuration using the DHCP client, use the command:

```
awplus(config-if)# ip address dhcp [client-id <interface>] [hostname <hostname>]
```

- If an IP interface is configured to get its IP address and subnet mask from DHCP, the interface does not take part in IP routing until the IP address and subnet mask have been set by DHCP.
- If you need to make a static entry in the DHCP server from which the device is obtaining its IP address, you need your device’s MAC address, which you can display by using the command: `show interface`

See the [DHCP Feature Overview and Configuration Guide](#) for more information about DHCP.
Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP) is used by your device to dynamically learn the mapping between the Layer 2 addresses and IP addresses of devices in the networks to which it is connected. Most hosts also have a MAC physical address in addition to the assigned IP address. For Ethernet, this is a 6-byte, globally unique number. ARP enables your device to learn the physical address of the host that has a given IP address.

When your device needs to forward packets to a destination whose Layer 2 address it does not know, it broadcasts an ARP request to determine the Layer 2 destination address to put on the packet. The ARP request is a broadcast packet and includes the target IP address. All stations on the LAN receive this broadcast but only one host recognizes its own IP address. It replies, thereby giving your device its physical address.

Your device creates a dynamic ARP entry in its ARP cache, to record the IP address to physical address mapping (also called a binding). It uses that ARP entry to find that host’s physical address when forwarding further packets to that address.

The ARP protocol is described in RFC 826, An Ethernet Address Resolution Protocol—or—Converting Network Protocol Addresses to 48 bit Ethernet Address for Transmission on Ethernet Hardware.

Static ARP entries

If your LAN includes hosts that do not support ARP, you can add a static ARP entry to the cache. However, it is rarely necessary to add an ARP entry this way. To add a static ARP entry, use the command:

```
awplus(config)# arp <ip-addr> <mac-address> [<port-number>] [alias]
```

or, if you have VRF-lite enabled:

```
awplus(config)# arp [vrf <vrf-name>] <ip-addr> <mac-address> [<port-number>] [alias]
```

Timing out ARP entries

Your device times out dynamic ARP entries to ensure that the cache does not fill with entries for hosts that are no longer active. If your device stops receiving traffic for a device specified in a dynamic ARP entry, it deletes the ARP entry after a configurable timeout period. Static ARP entries are not aged or automatically deleted.

Increasing the ARP timeout reduces the amount of network traffic. Decreasing the timeout makes your device more responsive to changes in network topology.

To set a timeout period, enter the interface mode, then use the command:

```
awplus(config-if)# arp-aging-timeout <0-432000>
```
Deleting ARP entries

To remove a static ARP entry, use the command:

```
awplus(config)# no arp <ip-addr>
```

or, if you have enabled VRF-lite:

```
awplus(config)# no arp [vrf <vrf-name>] <ip-addr>
```

To clear the ARP cache of dynamic entries, use the command:

```
awplus# clear arp-cache
```

This removes the dynamic ARP entries for all interfaces.

To display the entries in the ARP cache, use the command:

```
awplus# show arp
```

The ARP cache will be repopulated by the normal ARP learning mechanism. As long as the entries are relearned quickly enough, deleting dynamic ARP entries does not affect:

- routes
- OSPF neighbor status
- BGP peer status
- the TCP/UDP connection status
- VRRP status

Proxy ARP

Proxy ARP (defined in RFC 1027) deals with the situation where hosts in one subnet are sending ARP requests for IP addresses that are in a different subnet. Typically, this happens when the subnet mask configured on the requesting hosts does not match the subnet mask that has actually been allocated to their subnet.

Your device intercepts these ARP broadcast packets that are requesting IP addresses that are outside the local subnet, and substitutes its own physical address for that of the remote host. This occurs only if your device has the best route to the remote host.

By responding to the ARP request, your device is effectively saying to the requesting host 'send that traffic to me, and I will ensure it gets to that requested destination'. So, that subsequent packets from the local host, destined for the IP address outside the local subnet, are directed to your device's physical address, and it can then forward these to the remote host. The process is symmetrical.
Proxy ARP is disabled by default. To enable proxy ARP on an interface, use the commands:

```
awplus# interface <interface>
awplus(config-if)# ip proxy-arp
```

To disable Proxy ARP on an interface, use the command:

```
awplus(config-if)# no ip proxy-arp
```

To check Proxy ARP is enabled on an interface, use the `show running-config` command. If Proxy ARP has been enabled an entry shows `ip proxy-arp` below the interface it is enabled on. No `ip proxy-arp` entry below an interface in the config indicates Proxy ARP is disabled on that interface.

See the sample configuration commands and validation command with resulting output showing proxy ARP **enabled** on VLAN 2 below:

```
awplus#configure terminal
awplus(config)#interface vlan2
awplus(config-if)#ip proxy-arp
awplus(config-if)#end
awplus(config)#exit
awplus#show running-config
! interface vlan2
  ip proxy-arp
  ip address 192.168.2.2/24
!
```

See the sample configuration commands and validation command with resulting output showing proxy ARP **disabled** on VLAN 2 below:

```
awplus#configure terminal
awplus(config)#interface vlan2
awplus(config-if)#no ip proxy-arp
awplus(config-if)#end
awplus(config)#exit
awplus#show running-config
! interface vlan2
  ip address 192.168.2.2/24
!
```

**Local Proxy ARP**

Local Proxy ARP lets you stop MAC address resolution between hosts within an interface’s subnet. This ensures that traffic between hosts in an environment where hosts are isolated from each other (e.g. a Private VLAN) is directed through one forwarding point. This lets you monitor, filter, and control traffic between devices in the same subnet.

Local Proxy ARP extends proxy ARP by intercepting and responding to ARP requests between hosts within a subnet. Local proxy ARP responds to ARP requests with your
device’s own MAC address details instead of those from the destination host. This stops hosts from learning the MAC address of other hosts within its subnet.

When Local Proxy ARP is operating on an interface, your device does not generate or forward any ICMP-Redirect messages on that interface.

Local Proxy ARP is disabled by default. To enable local proxy ARP on an interface, use the commands:

```
awplus# interface <interface>
awplus(config-if)# ip local-proxy-arp
```

To disable local proxy ARP on an interface, use the command:

```
awplus(config-if)# no ip local-proxy-arp
```

To check Local Proxy ARP is enabled on an interface, use the `show running-config` command. If Local Proxy ARP has been enabled an entry shows `ip local-proxy-arp` below the interface it is enabled on. If there is no `ip local-proxy-arp` entry below an interface in the config, that indicates Local Proxy ARP is disabled on it.

See the sample configuration commands and validation command with resulting output showing local proxy ARP enabled on VLAN 1 below:

```
awplus#configure terminal
awplus(config)#interface vlan1
awplus(config-if)#ip local-proxy-arp
awplus(config-if)#end
awplus(config)#exit
awplus#show running-config
!
interface vlan1
  ip local-proxy-arp
  ip address 192.168.1.2/24
!
```

See the sample configuration commands and validation command with resulting output showing Local Proxy ARP disabled on VLAN 1 below:

```
awplus#configure terminal
awplus(config)#interface vlan1
awplus(config-if)#no ip local-proxy-arp
awplus(config-if)#end
awplus(config)#exit
awplus#show running-config
!
interface vlan1
  ip address 192.168.1.2/24
!
```
ARP logging

You can enable your device to log events that happen in the ARP cache, like the adding and deleting of static and dynamic ARP entries, and you can select either default hexadecimal notation (HHHH.HHHH.HHHH) or standard IEEE format hexadecimal notation (HH-HH-HH-HH-HH-HH) for the MAC addresses displayed in the ARP log output.

If this feature is enabled, ARP log messages are stored on the device in RAM. If the device is rebooted the ARP log messages are lost. ARP logging is disabled by default.

To enable ARP logging, use the command:

```
awplus(config)# arp log [mac-address-format ieee]
```

You can specify whether the MAC address is displayed in the default hexadecimal notation HHHH.HHHH.HHHH or in the standard IEEE format HH-HH-HH-HH-HH-HH.

To disable ARP logging, use the command:

```
awplus(config)# no arp log [mac-address-format ieee]
```

To display the ARP log messages, use the command:

```
awplus(config)# show log | include ARP_LOG
```

See the sample ARP log output and descriptions of the fields displayed in the sample ARP log output in the `arp log` command.

Domain Name System (DNS)

The Domain Name System allows you to access remote systems by entering human-readable device host names rather than IP addresses. DNS works by creating a mapping between a domain name, such as “www.alliedtelesis.com”, and its IP address. These mappings are held on DNS servers. DNS translates meaningful domain names into IP addresses for networking equipment to locate and address these devices. The benefits of DNS are that domain names:

- can map to a new IP address if the host's IP address changes
- are easier to remember than an IP address
- allow organizations to use a domain name hierarchy that is independent of any IP address assignment

AlliedWare Plus has the ability to resolve domain names for internally generated commands (DNS Client) as well as providing the DNS information to connected hosts (via DNS Relay and DHCP Server. The DNS Client is enabled automatically when at least one DNS server is configured on the device, with the command `ip name-server <ip-addr>`.
This client allows you to use domain names instead of IP addresses when using commands on your device, like ping, SSH, and copy.

The DNS Relay provides the presence of a local virtual DNS server which can service DNS lookup requests sent to it from local hosts. The DHCP Server can be configured to provide DNS information to DHCP clients during the lease process.

**Domain name parts**

Domain names are made up of a hierarchy of two or more name segments. Each segment is separated by a period. The format of domain names is the same as the host portion of a URL (Uniform Resource Locator). The first segment from the left is unique to the host, with each following segment mapping the host in the domain name hierarchy. The segment on the far right is a top-level domain name shared by many hosts.

**Server hierarchy**

A network of domain name servers maintains the mappings between domain names and their IP addresses. This network operates in a hierarchy that is similar to the structure of the domain names. When a local DNS server cannot resolve your request it sends the request to a higher level DNS server.

For example, to access the site "alliedtelesis.com", your PC sends a DNS inquiry to its local DNS server asking for the IP address matching alliedtelesis.com. If this address is already locally cached (following its recent use), the DNS server returns the IP address that matches alliedtelesis.com. If the DNS server does not have this address cached, it forwards the request upwards through the hierarchy of DNS servers until a DNS server can resolve the mapping. This means an often-used domain name is resolved quickly, while an uncommon or nonexistent domain may take longer to resolve or fail.

As well as the hierarchy of domain name servers accessible through the Internet, you can operate your own DNS server to map to private IP addresses within your network.

The DHCP server IP address can be either statically defined, or can be dynamically assigned via DHCPv4 option 6 using the `ip name-server` command and DHCP option 15 using the `ip domain-name` command if the DHCP client is configured.

See the DHCP Feature Overview and Configuration Guide for more information about DHCP and DHCP options.
DNS client

Your AlliedWare Plus device has a DNS Client that is enabled automatically when you configure a Name server address on to your device. This client allows you to use domain names instead of IP addresses when using commands on your device.

To add a DNS server to the list of servers that the device sends DNS queries to, use the command:

```
awplus(config)# ip name-server <ip-addr>
```

To check the list of servers that the device sends DNS queries to, use the command:

```
awplus# show ip name-server
```

To add a default domain name used to append to DNS requests, use the command:

```
awplus(config)# ip domain-name <domain-name>
```

For example, to use DNS to match hostnames to your internal network “example.net”, use the command:

```
awplus(config)# ip domain-name example.net
```

If you then use the command **ping host2**, your device sends a DNS request for host2.example.net. To check the domain name configured with this command, use the command:

```
awplus# show ip domain-name
```

Alternatively you can create a list of domain names that your device will try in turn by using the command:

```
awplus(config)# ip domain-list <domain-name>
```

For example, to use DNS to match incomplete hostnames to the top level domains “.com”, and “.net”, use the commands:

```
awplus(config)# ip domain-list .com
awplus(config)# ip domain-list .net
```

If you then use the command **ping alliedtelesis**, your device sends a DNS request for alliedtelesis.com and if no match was found your device would then try alliedtelesis.net.

To check the entries in the domain list, use the command:

```
awplus# show ip domain-list
```

To disable the DNS client on your device, use the command:

```
awplus(config)# no ip domain-lookup
```

To check the status of the DNS client on your device, and the configured servers and domain names, use the command:

```
awplus# show hosts
```
DNS Relay

Enabling DNS Relay your switch provides the capability for it to act as a local virtual DNS server. It can then service DNS lookup requests sent to it from local hosts. Acting as a DNS Relay, the switch will usually relay the requests to an external, or upstream, DNS server. By default, DNS Relay is disabled.

**Note:** When running VRF-lite, the DNS Relay functions will apply separately within each VRF Instance.

Optionally, DNS name resolver caching may be enabled on the DNS Relay, which can provide some lookup speed advantage and avoid unnecessary repeated requests to external DNS servers. By default, DNS caching is disabled.

When the DNS Relay name resolver cache is enabled on your switch, the switch will maintain a cache of recently used mappings between domain names and IP addresses so that other identical requests can be responded to without further reference to an external, or upstream DNS server. When the switch receives a DNS query from a client the switch will attempt to match the request with entries in this cache. If the switch does not have this address cached, it forwards the request upwards through the hierarchy of DNS servers for resolution. The DNS cache has a limited size, and times out entries after a specified period of up to 60 minutes.

The relaying of DNS queries is required for use in networks where the network administrator wishes to have the flexibility to easily change to using a different DNS server. By configuring just the gateway device with the actual DNS address(es), and configuring all other devices to send their DNS requests to the gateway device, the administrator need only update the DNS address(es) in one place (on the gateway device) when changing to a different DNS server. This is far more convenient than having to update DNS addresses in all the individual hosts in the network.

DNS Relay uses the DNS server list configured by the `ip name-server` command to forward DNS query packets. To enable DNS Relay you need to configure the list of servers that the device sends DNS queries to and then enable DNS forwarding, as shown in the following example for a DNS server with an IPv4 address:

```
awplus# configure terminal
awplus(config)# ip name-server 192.168.1.1
awplus(config)# ip name-server 192.168.1.2
awplus(config)# ip dns forwarding
```

DNS Relay requires that IP domain lookup is enabled. IP domain lookup is enabled by default, but if it has been disabled, you can re-enable it by using the command:

```
awplus(config)# ip domain-lookup
```

Note both IPv4 and IPv6 support DNS record types. IPv4 and IPv6 are supported in DNS name-to-address and DNS address-to-name lookup processes. Specifying a name server and enabling DNS forwarding maps both IPv4 and IPv6 addresses.
You can configure DNS Relay to use IPv6 addresses using the same commands used to configure DNS Relay to use IPv4 addresses, as shown in the following example:

```
awplus# configure terminal
awplus(config)# ip name-server 2001:0db8:010d::1
awplus(config)# ip name-server 2001:0db8:010d::2
awplus(config)# ip dns forwarding
```

You can then configure DNS Relay behavior with the following commands:

To set the number of times a switch will retry to forward DNS queries, use the command:

```
awplus(config)# ip dns forwarding retry <0-100>
```

To set the number of seconds to wait for a response, use the command:

```
awplus(config)# ip dns forwarding timeout <0-3600>
```

To set the DNS forwarding dead-time period in seconds, use the command:

```
awplus(config)# ip dns forwarding dead-time <60-43200>
```

At the dead-time period set, the switch stops sending requests to an unresponsive server.

To set the interface to use for forwarding and receiving DNS queries, use the command:

```
awplus(config)# ip dns forwarding source-interface <interface-name>
```

To specify the DNS Relay name resolver cache size and lifetime, use the command:

```
awplus(config)# ip dns forwarding cache [size <0-1000>] [timeout <60-3600>]
```

To remove entries from the DNS Relay name resolver cache, use the command:

```
awplus(config)# clear ip dns forwarding cache
```

Information which may be useful for troubleshooting DNS Relay is available using the DNS Relay debugging function. To enable DNS Relay debugging, use the command:

```
awplus# debug ip dns forwarding
```

To display the status of DNS Relay, use the command:

```
awplus# show ip dns forwarding
```

To display the status of DNS Relay name servers, use the command:

```
awplus# show ip dns forwarding server
```

To display the DNS Relay name resolver cache, use the command:

```
awplus# show ip dns forwarding cache
```
DNS domain name matching

DNS domain name matching allows you to specify a domain name suffix to match on when a client does a DNS lookup. When a match is detected the appropriate name server is used for resolving the address. For example, you might want to do the following:

- For all general DNS lookups such as www.Example.com, use name server 1.2.3.4.
- For internal domain names such as atlnz.lc, use the name server received on interface ppp1.

Domain matching order

The matching procedure used for this feature is a tail-first match against the requested domain in the DNS request packet, and the longest specified domain will match first. The following examples show you how this works:

If the configuration is set up so that:

- "com" will be sent to DNS server 10.1.1.1.
- "example.com" will be sent to DNS server 20.1.1.1.
- otherwise everything else will be sent to the primary DNS server 30.1.1.1.

Then the following will occur:

1. When a DNS request for "mail.example.com" is received, the longest successful match is "example.com", so the request will be forwarded on to DNS server 20.1.1.1.
2. When a DNS request for "something.com" is received, the longest successful match is against "com", so the request will be forwarded on to DNS server 10.1.1.1.
3. When a DNS request for "something_else.org" is received, there is no match against the specified domains, so the request will be forwarded on to the primary DNS server 30.1.1.1.

Note that this feature will only work for DNS lookups from a down-stream host that are relayed via the device, which is enabled by using the `ip dns forwarding` command. The feature functionality does not apply to DNS lookups originating from the device itself, for example "ping hostname" executed on the device.

The following example shows you how to create a domain list that can be used as a suffix-list for DNS lookups. The domain list includes domains that are internal to the company such as “engineering.acme” or “intranet.acme”.

To create a domain list called “corporatedomains”, use the command:

```
awplus(config)# ip dns forwarding domain-list corporatedomains
```

To give a description to the domain list, use the command:
awplus(config-domain-list)# description Our internal network domains

To add a domain to the list, use the command:

awplus(config-domain-list)# domain engineering.acme

awplus(config-domain-list)# domain intranet.acme
DNS operation with VRF-lite

On devices that support VRF-lite, you can configure DNS Relay functionality to be VRF aware. In this mode DNS Relay messages can be forwarded within specified VRF instances. VRF aware DNS services to remotely connected DNS clients is also supported. These VRF aware services include: ping, traceroute, telnet client, SSH client, and tcpdump.

Configuring DNS operation with VRF-lite

The `ip name-server [vrf <name>] <ip-addr>` command configures a name-server for the specified VRF. This command assigns the address of one or more name servers to a VRF table to be used for name and address resolution. If no VRF-lite instance (vrf<name>) is specified, the name-server is configured for the global VRF. A name-server that is configured on the global VRF will apply to both the DNS relay and DNS client. Note that the DNS client is not VRF aware. A VRF specific name-cache is created within the DNS relay for every VRF instance that has a name-server configured.

A maximum of three name-servers may be defined for each DNS-relay instance.

The configuration command, `ip dns forwarding`, will apply to all VRF instances configured on the device and not on a per VRF basis.

The configuration commands listed below apply to all VRF instances configured on the device and not on a per VRF basis. Timeouts are in seconds as per existing commands:

- `ip dns forwarding retry`
- `ip dns forwarding timeout`
- `ip dns forwarding dead-time`
- `ip dns forwarding source-interface`
- `ip dns forwarding cache`

The following `show` commands provide output information for the VRF instance specified. If a VRF instance is not specified, output is shown for all VRF instances, including the global instance and the output will be formatted in a way that distinguishes the information for each VRF.

- `show ip dns [vrf <name>|global] forwarding server`
- `show ip dns [vrf <name>|global] forwarding cache`
- `show ip name-server [vrf <name>|global]`

The DNS cache can also be cleared on a per VRF instance basis by using the `clear ip dns [vrf <name>|global] forwarding cache` command.
The following commands show how to configure a DNS relay name-server for both the specified VRF instance VRF red, and the global VRF instance.

To configure a DNS relay name-server for the VRF-lite instance red:

```
awplus# configure terminal
awplus(config)# ip name-server vrf red 192.168.0.1
awplus(config)# ip domain-lookup
```

To configure a DNS relay name-server for the global VRF instance:

```
awplus# configure terminal
awplus(config)# ip name-server 192.168.1.1
awplus(config)# ip domain-lookup
```

**DHCP options**

When your device is using its DHCP client for an interface, it can receive the following DHCP options from the DHCP server:

- Option 6 - a list of DNS servers. This list appends to the DNS servers set on your device with the `ip name-server` command.

- Option 15 - a domain name used to resolve host names. This option replaces the domain name set with the `ip domain-name` command.

See the DHCP Feature Overview and Configuration Guide for more information about DHCP and DHCP options.

**DHCP operation with VRF-lite**

On devices that support VRF-lite, you can configure DHCP Relay to forward packets within a VRF-lite instance.

DHCP messages between DHCP Clients and a DHCP Server are then able to be relayed between VLAN interfaces within a VRF-lite instance.

When an DHCP Relay instance is enabled on a VLAN interface, using the command: `ip dhcp-relay server-address`, the relay will be placed within the same VRF-lite instance that the VLAN belongs to. If the VLAN does not belong to a VRF-lite instance, then the DHCP Relay will be placed in the default (global) VRF.

Note that DHCP option 82 is still supported on a DHCP Relay operating within a VRF-lite instance.

The following commands can be used to monitor DHCP relay functions:

- `show counter dhcp-relay [vrf <vrf-name>] [global]`
- `show ip dhcp-relay [interface <interface-name>]`
**Example**

The following example shows how to configure DHCP relay for a VLAN interface that is associated with named VRF-lite instance VRF red:

```
awplus# configure terminal
awplus(config)# interface vlan1
awplus(config-if)# ip vrf red
awplus(config-if)# ip address 192.168.0.1/24
awplus(config-if)# ip dhcp-relay server-address 192.168.1.1
```

**Internet Control Message Protocol (ICMP)**

The Internet Control Message Protocol (ICMP) allows networking devices to send information and control messages to other devices or hosts. Your device implements all non-obsolete ICMP functions.

The following table lists the ICMP messages implemented by your device.

<table>
<thead>
<tr>
<th>ICMP MESSAGE TYPE</th>
<th>DEVICE RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Echo reply (0)</td>
<td>This is used to implement the ping command. Your device sends out an echo reply in response to an echo request.</td>
</tr>
<tr>
<td>Destination unreachable (3)</td>
<td>This message is sent when your device drops a packet because it did not have a route to the destination.</td>
</tr>
<tr>
<td>Redirect (5)</td>
<td>Your device issues this message to inform a local host that its target is located on the same LAN (no routing is required) or when it detects a host using a non-optimal route (usually because a link has failed or changed its status). For example, if your device receives a packet destined to its own MAC address, but with a destination IP address of another host in the local subnet, it returns an ICMP redirect to the originating host. ICMP redirects are disabled on interfaces on which local proxy ARP is enabled.</td>
</tr>
<tr>
<td>Echo request (8)</td>
<td>This is related to echo replies. If your device receives an echo request, it sends an echo reply. If you enter the ping command, your device generates echo requests.</td>
</tr>
<tr>
<td>Router Advertisements (10)</td>
<td>These are Router Discovery Protocol messages. If Router Discovery is enabled, your device sends these to announce the IP addresses of the sending interface.</td>
</tr>
<tr>
<td>Time to Live Exceeded (11)</td>
<td>If the TTL field in a packet falls to zero, your device sends this message. This occurs when there are too many hops in the path that a packet is traversing.</td>
</tr>
</tbody>
</table>
ICMP messages are enabled on all interfaces by default. You can control the flow of ICMP messages across different interfaces using the access-list commands. See the following sections in your product’s Command Reference, available on alliedtelesis.com:
- IPv4 Hardware Access Control List (ACL) Commands
- IPv4 Software Access Control List (ACL) Commands

ICMP Router Discovery Protocol (IRDP)

Router Discovery

Some AlliedWare Plus devices support the router specification sections of IRDP (RFC 1256, ICMP Router Discovery Messages). If this feature is configured, your device sends router advertisements periodically and in response to router solicitations. It does not support the Host Specification section of this RFC.

Benefits

Before an IP host can send an IP packet, the host has to know the IP address of a neighboring router that can forward the packet to its destination. ICMP Router Discovery messages let routers automatically advertise themselves to hosts. Other methods either require someone to manually keep these addresses current, or require DHCP to send router addresses.

Router Discovery process

The following table summarizes what happens when Router Discovery advertisements are enabled on an interface.

<table>
<thead>
<tr>
<th>WHEN...</th>
<th>THEN...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router Discovery advertising starts on an interface because:</td>
<td>Your device multicasts a router advertisement and continues to multicast them periodically until router advertising is disabled.</td>
</tr>
<tr>
<td>■ your device starts up, or</td>
<td></td>
</tr>
<tr>
<td>■ you enable advertisements on your device or on an interface.</td>
<td></td>
</tr>
<tr>
<td>A host starts up.</td>
<td>The host may send a router solicitation message.</td>
</tr>
<tr>
<td>Your device receives a router solicitation.</td>
<td>Your device multicasts an early router advertisement from the interface on which it received the router solicitation.</td>
</tr>
<tr>
<td>A host receives a router advertisement.</td>
<td>The host stores the IP address and preference level for the advertisement lifetime.</td>
</tr>
<tr>
<td>The lifetimes of all existing router advertisements on a host expire.</td>
<td>The host sends a router solicitation.</td>
</tr>
</tbody>
</table>
Advertisement messages

A router advertisement is an ICMP (type 10) message that contains the following:

- in the destination address field of the IP header, the interface’s configured advertisement address, either 224.0.0.1 or 255.255.255.255.
- in the lifetime field, the interface’s configured advertisement lifetime.
- in the Router Address and Preference Level fields, the addresses and preference levels of all the logical interfaces that are set to advertise.

Your device does not send router advertisements by default.

Solicitation message

A router solicitation is an ICMP (type 10) message containing:

- **source address**: an IP address belonging to the interface from which the message is sent
- **destination address**: the configured Solicitation Address, and
- **Time-to-Live**: 1 if the Destination Address is an IP multicast address; at least 1 otherwise.

Advertisement interval

The router advertisement interval is the time between router advertisements. For the first few advertisements sent from an interface (up to 3), your device sends the router advertisements at intervals of at most 16 seconds. After these initial transmissions, it sends router advertisements at random intervals between the minimum and maximum intervals that the user configures, to reduce the probability of synchronization with the advertisements from other routers on the same link. By default, the minimum is 450 seconds (7.5 minutes), and the maximum is 600 seconds (10 minutes).
Preference level
The preference level is the preference of the advertised address as a default router address relative to other router addresses on the same subnet. By default, all routers and Layer 3 switches have the same preference level, zero. While it is entered as a decimal from 0 to 2147483647, it is encoded in router advertisements as a two’s-complement hex integer from 0x8000000 to 0x7fffffff. A higher preference level is preferred over a lower value.

Lifetime
The lifetime of a router advertisement is how long the information in the advertisement is valid. By default, the lifetime of all advertisements is 1800 seconds (30 minutes).

Address type
Your device can send its router advertisements using either a broadcast or multicast destination address. By default, your device sends router advertisements using the all-systems multicast address (224.0.0.1). However, on networks where the hosts do not support IP multicast you must use the broadcast address (255.255.255.255). To change the address type to broadcast on an interface, use the command:

```
awplus(config-if)# ip irdp broadcast
```

To change the address type back to multicast, use the `no` variant of the above command, or use the command:

```
awplus(config-if)# ip irdp multicast
```

Configuration procedure
Perform the following to configure your device to send router advertisements:

**Step 1: Enter the interface to advertise.**

Enter the configuration mode for the interface, using the command:

```
awplus(config)# interface <interface>
```

**Step 2: Change the address type.**

By default, your device sends router advertisements using a multicast destination address. If hosts on your network do not support this, change the address type to broadcast, using the command:

```
awplus(config-if)# ip irdp broadcast
```

**Step 3: Configure the advertisement interval and lifetime.**

By default, your device sends router advertisements every 7.5 to 10 minutes, with a lifetime of 30 minutes. These settings are likely to work well in most situations, and will not cause a large amount of extra traffic, even if there are several routers on the LAN. If you change these settings, keep the following proportions:

```
lifetime=3 x maxadvertisementinterval
minadvertisementinverval=0.75 x maxadvertisementinterval
```
You cannot set the maximum advertisement interval below the minimum interval. If you are lowering the maximum interval to a value below the current minimum interval, you must change the minimum value first. This also applies to changing the minimum interval above the current maximum interval.

To change the maximum advertisement interval, use the command:

```
awplus(config-if)# ip irdp maxadvertinterval <4-1800>
```

To change the minimum advertisement interval, use the command:

```
awplus(config-if)# ip irdp minadvertinterval <3-1800>
```

To change the lifetime for your device’s router advertisements, use the command:

```
awplus(config-if)# ip irdp lifetime <0-9000>
```

**Step 4: Set preference levels.**

By default, every interface has the same preference for becoming a default router. To give the interface a higher preference, increase the preference level. To give it a lower preference, decrease this value.

To set the preference level for all addresses on this interface, use the command:

```
awplus(config-if)# ip irdp preference <0-2147483647>
```

To set the preference for a specific address on the interface, use the command:

```
awplus(config-if)# ip irdp address <ip-address> preference <0-2147483647>
```

**Step 5: Enable advertising on the interface.**

To enable router advertisements on an interface, enter the interface mode and use the command:

```
awplus(config-if)# ip irdp
```

**Step 6: Enable advertising on your device.**

To globally enable router advertisements on your device, enter global configuration mode and use the command:

```
awplus(config-if)# exit
awplus(config)# router ip irdp
```

**Step 7: Check advertise settings.**

To view the IRDP configuration on the interface, use the command:

```
awplus(config)# exit
awplus# show ip irdp interface [<interface-name>]
```

To view the global IRDP configuration for your device, use the command:

```
awplus# show ip irdp
```

**Debugging IRDP**

Information which may be useful for troubleshooting IRDP is available using the IRDP debugging function. To enable IRDP debugging, use the command:

```
awplus# debug ip irdp {event|nsm|receive|send|both|detail|all}
```
Checking IP Connections

To verify connections between networks and network devices, use the ping (Packet Internet Groper) and trace route functions on your device.

Ping

Ping tests the connectivity between two network devices to determine whether each network device can 'see' the other device. Echo request packets are sent to the destination addresses and responses are displayed on the console.

If you can ping the end destination, then the physical, Layer 2 and Layer 3 links are functioning, and any difficulties are in the network or higher layers.

If pinging the end destination fails, use traceroute to discover the point of failure in the route to the destination.

To ping a device, use the command:

awplus# ping {<hostname>|<ipaddr>}

Traceroute

You can use traceroute to discover the route that packets traverse between two systems running the IP protocol. Traceroute sends an initial UDP packet with the Time To Live (TTL) field in the IP header set to a starting value of 1. The TTL field is increased by one for every subsequent packet sent until the destination is reached. Each hop along the path between two systems responds with a TTL exceeded packet (ICMP type 11) and from this the path is determined.

To use traceroute, use the command:

awplus# traceroute {<hostname>|<ipaddr>}

or, if you have enabled VRF-lite:

awplus# traceroute [vrf <vrf-name>] {<hostname>|<ipaddr>}

Enter either the hostname or the IP address of the device you are trying to reach.
IP Helper (UDP Broadcast Helper)

On switches that support it, the IP Helper feature allows the switch to receive UDP broadcasts on one subnet, and forward them as broadcasts or unicast into another subnet, so a client can use an application which uses UDP broadcast (such as Net-BIOS) when the client and server are located in different subnets. The IP Helper feature forwards UDP broadcast network traffic to specific hosts on another subnet and/or to the broadcast address of another subnet.

When the IP Helper feature is enabled on a VLAN interface, the UDP broadcast packets received on the interface are processed for forwarding out through another interface into another subnet. Depending on the nature of the ip-helper addresses configured, the UDP broadcasts will be unicast forwarded to a single host in the destination subnet, or unicast forwarded to multiple hosts in the destination subnet, or broadcast to the broadcast address of the destination subnet. Not all UDP broadcasts will be forwarded when IP Helper is configured. The set of broadcasts to be forwarded can be defined by specifying the destination UDP port(s) of the packets you wish to forward.

The command to enable the forwarding of UDP broadcasts received on a given interface is `ip helper-address` (entered in interface configuration mode). The `ip forward-protocol udp` command specifies types of broadcast packets to forward.

Multiple different destination addresses can be specified by using multiple instances of the `ip helper-address` command under the same interface. If a destination address is specified that is actually the broadcast address of one of the subnets directly connected to the switch, then the UDP packets will be forwarded as broadcasts onto that subnet.

Likewise, multiple different types of UDP packet can be specified for forwarding by specifying multiple different destination ports using the `ip forward-protocol udp` command.

**Note:** The types of UDP broadcast packets that the switch will forward are only those specified by the `ip forward-protocol` command(s). The IP Helper process does not forward any other UDP packet types by default.
IP Directed Broadcast

IP directed-broadcast is enabled and disabled per VLAN interface. When enabled, a directed broadcast packet is forwarded to an enabled VLAN interface if received on another subnet.

An IP directed broadcast is an IP packet whose destination address is a broadcast address for some IP subnet, but originates from a node that is not itself part of that destination subnet. When a directed broadcast packet reaches a switch that is directly connected to its destination subnet, and IP directed-broadcast is enabled on the interface via which the switch connects to that destination subnet, the packet is flooded as a broadcast on the destination subnet.

The `ip directed-broadcast` command only controls the flooding of directed broadcasts when they reach target subnets. The command affects the final transmission of the directed broadcast on its destination subnet. It does not affect the transit unicast routing of IP directed broadcasts.

If the `no ip directed-broadcast` command is configured for an interface, directed broadcasts destined for the subnet where the interface is attached will be dropped instead of broadcast.

Support for Network Load Balancing (NLB)

Introduction

NLB is one of the clustering technologies available from Microsoft®. It provides high availability for services such as HTTP and FTP, by grouping identical servers into a cluster and sharing the network load between all currently-available servers in the cluster.

This section only applies to SBx8100, SBx908, and x900 switches.

Network Load Balancing clustering modes

There are two modes for Network Load Balancing with Windows 2003®: unicast and multicast. Multicast mode has a further option of IGMP Multicasting. This section discusses how these modes work.

Note that clustering only works, in multicast or unicast mode, if all packets sent to the cluster’s IP address are sent to all nodes in the cluster. This means that the cluster-side switch must flood traffic to all ports that are connected to the members of the cluster.
**Unicast mode**

In **unicast mode**, all hosts in the cluster share a single unicast ‘cluster’ MAC address, to go with the cluster IP address. This mode has the disadvantage that it stops cluster hosts from using their own ‘burned-in’ MAC addresses, so hosts cannot contact each other unless you install a second NIC card on each host and use that for intra-cluster communication.

The cluster operation forces the switch to flood all packets that are destined for the cluster, by stopping the switch from learning the cluster MAC address. The switch cannot learn the cluster MAC address because it never appears in the source field of the Ethernet headers of packets sent from the clustered servers.

Instead, each NIC uses a special unicast MAC address in the source field of the Ethernet header. The cluster MAC address must never be found in the source field of the Ethernet header, because otherwise the switch will learn the cluster MAC address, and stop flooding packets to all ports, and only one node in the cluster will receive traffic for the cluster’s IP.

The following diagrams illustrate unicast mode:
Client Workstation
IP address: 172.16.0.40
MAC address: 00-00-54-1A-71-B3

Clustered Servers
Cluster virtual IP: 172.16.0.127
Cluster virtual MAC: 02-BF-AC-10-00-7F

ARPs
Source MAC address: 02-01-AC-10-00-7F
Dest MAC address: 00-00-54-1A-71-B3
Sender Hardware address: 02-BF-AC-10-00-7F
Sender IP address: 172.16.0.127
Target HW address: 00-00-54-1A-71-B3
Target IP address: 172.16.0.40

These MAC addresses are different

IP          MAC
02-BF-AC-10-00-7F

Port          MAC
1 02-01-AC-10-00-7F
2 02-02-AC-10-00-7F
3 02-03-AC-10-00-7F
Clustered Servers
Cluster virtual IP: 172.16.0.127
Cluster virtual MAC: 02-BF-AC-10-00-7F

Client Workstation
IP address: 172.16.0.40
MAC address: 00-00-54-1A-71-B3

IP Packet is flooded to all of Ports 1, 2, 3 because the Dest MAC 02-BF-AC-10-00-7F is not in the switch’s FDB

FDB Table
<table>
<thead>
<tr>
<th>Port</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>02-01-AC-10-00-7F</td>
</tr>
<tr>
<td>2</td>
<td>02-02-AC-10-00-7F</td>
</tr>
<tr>
<td>3</td>
<td>02-03-AC-10-00-7F</td>
</tr>
</tbody>
</table>

ARP Table
<table>
<thead>
<tr>
<th>IP</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.16.0.127</td>
<td>02-BF-AC-10-00-7F</td>
</tr>
</tbody>
</table>

Network Load Balancing clustering modes | Page 27
Multicast mode

In **multicast mode**, hosts in the cluster use their real ‘burned-in’ MAC address in the source field of the Ethernet header. However, they answer ARP requests for the cluster IP address with a multicast MAC in the ARP packet’s payload, while the Ethernet header on the ARP reply still has the real MAC address. This allows cluster hosts to contact each other.

In both multicast and unicast mode the mechanism to cause flooding is as such:

- ARP requests for the cluster IP are answered with the virtual cluster MAC address in the ARP packet’s payload.
- The node must send all packets, including ARP requests and replies, with a different source MAC in the Ethernet header.
- This prevents the switch from entering the virtual cluster MAC into the forwarding database (FDB).

The following diagrams illustrate multicast mode:
Client Workstation
IP address: 172.16.0.40
MAC address: 00-00-54-1A-71-B3

Clustered Servers
Cluster virtual IP: 172.16.0.127
Cluster virtual MAC: 01-00-11-22-33-44

ARP response: 172.16.0.127 is at HW address 02-BF-AC-10-00-7F
Source MAC address: 02-01-AC-10-00-7F
Dest MAC address: 00-00-54-1A-71-B3
Sender Hardware address: 01-00-11-22-33-44
Sender IP address: 172.16.0.127
Target HW address: 02-BF-AC-10-00-7F
Target IP address: 172.16.0.40

IP                MAC
01-00-11-22-33-44
172.16.0.127

ARP Table
IP    MAC
172.16.0.127 01-00-11-22-33-44

FDB Table
Port  MAC
1     02-01-AC-10-00-7F
2     02-02-AC-10-00-7F
3     02-03-AC-10-00-7F
Multicast mode

Clustered Servers
Cluster virtual IP: 172.16.0.127
Cluster virtual MAC: 01-00-11-22-33-44

Client Workstation
IP address: 172.16.0.40
MAC address: 00-00-54-1A-71-B3

IP
MAC
01-00-11-22-33-44
172.16.0.127

ARP Table
IP Packet is forwarded to all of Ports 1, 2, 3 because it has a multicast Dest MAC, so the switch Multicasts it.

FDB Table
Port | MAC
--- | ---
1   | 02-01-AC-10-00-7F
2   | 02-02-AC-10-00-7F
3   | 02-03-AC-10-00-7F

Client Workstation
IP address: 172.16.0.40
MAC address: 00-00-54-1A-71-B3

IP Packet

Clustered Servers
Cluster virtual IP: 172.16.0.127
Cluster virtual MAC: 01-00-11-22-33-44

ARP Table
IP | MAC
--- | ---
172.16.0.127 | 01-00-11-22-33-44

FDB Table
Port | MAC
--- | ---
1   | 02-01-AC-10-00-7F
2   | 02-02-AC-10-00-7F
3   | 02-03-AC-10-00-7F
**Multicast mode with the IGMP option selected**

This mode uses IGMP (Internet Group Management Protocol) to prevent the switch from flooding all ports; instead traffic only goes to NLB ports.

This mode also causes the cluster to use its real ‘burned in’ MAC address in the source field of the Ethernet header. The cluster will answer ARP requests for the cluster IP address with a multicast MAC in the ARP packet’s payload, while the Ethernet header on the ARP reply still has the real MAC address. Note though, that the MAC address is slightly different to the previously discussed multicast mode—the MAC address starts with the bytes 01:00:5e, which still identifies it as an IP-multicast MAC address:

The Windows client will happily add this IP / MAC combination to its internal ARP table (on a Windows or Linux PC, use the command `arp-a` to see all ARP entries). All IP traffic destined for 172.16.0.127 is then sent with an Ethernet destination of 01:00:5e:7f:00:7f. The cluster node responds with its burned in MAC in the Ethernet source field.

Microsoft’s IGMP multicasting mode cleverly turns the concept of IGMP clients and hosts on its head. As far as an Allied Telesis switch is concerned, the clustered servers are seen as IGMP clients and the IGMP multicast data (stream) is sent by the workstations trying to access the cluster. To ensure that the switches do correctly forward the multicast data, the servers send IGMP reports for a group that corresponds to the MAC address they are putting in their ARP responses. Reports are sent frequently by the NLB servers so there is no concern about IGMP entries timing out on the switch.

The following diagrams illustrate multicast mode with the IGMP option selected:
Cluster virtual IP: 172.16.0.127
Cluster virtual MAC: 01-00-5E-7F-00-7F

Group            Ports
224.127.0.127
(01-00-5F-7F-00-7F)

Client Workstation
IP address: 172.16.0.40
MAC address: 00-00-54-1A-71-B3

Clustered Servers

Cluster virtual IP: 172.16.0.127
Cluster virtual MAC: 01-00-5E-7F-00-7F

IP                MAC
01-00-5E-7F-00-7F
172.16.0.127

ARP response: 172.16.0.127 is at
HW address 02-BF-AC-10-00-7F
Source MAC address: 02-01-AC-10-00-7F
Dest MAC address: 00-00-54-1A-71-B3
Sender Hardware address: 01-00-5E-7F-00-7F
Sender IP address: 172.16.0.127
Target HW address: 00-00-54-1A-71-B3
Target IP address: 172.16.0.40

Client Workstation
IP address: 172.16.0.40
MAC address: 00-00-54-1A-71-B3

ARP Table
IP          MAC
172.16.0.127 01-00-5E-7F-00-7F
Multicast mode with the IGMP option selected

Clustered Servers
- Cluster virtual IP: 172.16.0.127
- Cluster virtual MAC: 01-00-5E-7F-00-7F

Clustered Servers
- Cluster virtual IP: 172.16.0.127
- Cluster virtual MAC: 01-00-5E-7F-00-7F

Client Workstation
- IP address: 172.16.0.40
- MAC address: 00-00-54-1A-71-B3

IGMP snooping Table
- Group: 224.127.0.127
- Ports: 1, 2, 3

ARP Table
- IP: 172.16.0.127
- MAC: 01-00-5E-7F-00-7F

IP Packet is forwarded to all of Ports 1, 2, 3 because there is an IGMP join for 01-00-5E-7F-00-7F on each of those ports.
Support for NLB in x-Series Switches

The operation of the NLB modes is optimized for Layer 2 networks. That is, the modes all work smoothly if the client work-stations and clustered servers are all in the same IP subnet, and any switches in between them are just operating at Layer 2.

In the purely Layer 2 environment, it is the client workstation that learns the ARP entry for the cluster's virtual IP/virtual MAC, and then sends cluster-directed packets to the virtual MAC. The Layer 2 switches in between then behave as described in the illustrations above.

For a Layer 2 network, the requirements on the switches are simple. In fact, NLB has been designed to make use of standard behaviors of Layer 2 switches. Therefore, in this environment, the switches don't even need to be aware that NLB is being used in the network.

Allied Telesis x-series switches, like almost all switches in the market, provide good support for NLB when operating at Layer 2.

When the switches are operating at Layer 3, the situation is rather different. If the client workstations are in one subnet, and the clustered servers are in another subnet, and a switch in between is performing the Layer 3 routing between those subnets, then the switch needs to be explicitly NLB-aware.

The key point is that the switch needs to learn the ARP entries for the Cluster's virtual IP and Virtual MAC, and apply those ARP entries to multiple egress ports. It is not standard for a switch to apply the same ARP entry to multiple ports, and, as will be discussed below, each of the NLB modes required other non-standard behavior on the part of the Layer 3 switch.
**Unicast Mode**

When a switch operating at Layer 3 sits in front of a set of clustered servers operating in Unicast Mode, the ARP conversation is as below:

1. **Switch**
   - IP address: 172.16.0.23
   - MAC address: 00-00-CD-12-34-56

2. **Client Workstation**
   - IP address: 192.168.54.23
   - MAC address: 00-00-54-1A-71-B3

3. **Clustered Servers**
   - Cluster virtual IP: 172.16.0.127
   - Cluster virtual MAC: 02-BF-AC-10-00-7F

   **ARP request** - who has IP address 172.16.0.127?
   - Source MAC: 00-00-CD-12-34-56
   - Dest MAC: FF-FF-FF-FF-FF-FF
   - Sender Hardware address: 00-00-CD-12-34-56
   - Sender IP address: 172.16.0.23
   - Target HW address: 00-00-00-00-00-00
   - Target IP address: 172.16.0.127

   **Switch Layer-3 routes packets from one subnet to the other**

4. **Subnet 1**
   - Switch IP address: 172.16.0.23
   - MAC address: 00-00-CD-12-34-56

   **ARP**

5. **Clustered Servers**
   - Cluster virtual IP: 172.16.0.127
   - Cluster virtual MAC: 02-BF-AC-10-00-7F

   **ARP response**: 172.16.0.127 is at HW address 02-BF-AC-10-00-7F
   - Source MAC address: 02-01-AC-10-00-7F
   - Dest MAC address: 00-00-CD-12-34-56
   - Sender Hardware address: 02-BF-AC-10-00-7F
   - Sender IP address: 172.16.0.127
   - Target HW address: 00-00-CD-12-34-56
   - Target IP address: 172.16.0.23

6. **Subnet 2**
   - Client Workstation IP address: 192.168.54.23
   - MAC address: 00-00-54-1A-71-B3

   **Switch**
   - IP address: 172.16.0.23
   - MAC address: 00-00-CD-12-34-56
The ARP responses in which the Sender MAC address does not match the Source MAC in the Ethernet header is referred to as a **Disparate ARP**.

In Allied Telesis x-series switches, the default behavior upon receiving a Disparate ARP response is:

- An ARP entry is created, pointing to the IP/MAC address in the content of the ARP packet, with the egress port being the port on which the ARP packet was received.

- An FDB entry is created for the MAC address in the content of the ARP packet, associated with the port on which the ARP was received. This is despite the fact that the MAC address in question was not the source MAC in the Ethernet header of the ARP packet. This FDB entry is not created by normal MAC learning, but is specifically created when the ARP entry is created, so that the hardware forwarding process (which uses the FDB as the way to find the egress port) works correctly.

The problem, though, is that the ARP entry (and FDB entry) is only ever associated with one egress port. So, rather than associating itself with multiple egress ports, it will jump from port to port as the replies from multiple servers arrive in quick succession. This fails to provide the behavior that the cluster requires—whereby packets destined to the cluster are sent to all cluster members at once.

So, AlliedWare Plus provides a specific, NLB-friendly, mode of dealing with Disparate ARP responses. In this mode, the behavior upon receiving a disparate ARP response is:

- An ARP entry is created, pointing to the IP/MAC address in the content of the ARP packet, with no specific egress port - the ‘egress port’ is set to ‘flood’, which means that packets matching this ARP entry are flooded to all ports in the egress VLAN.

They appear in the output of the command **show ARP** as:

```
ATX8106M2#show arp
IP Address      MAC Address   Interface Port Type
192.168.200.34  001a.a004.25ef vlan200 flood dynamic
172.21.0.3      02bf.ac15.0005 vlan40  flood  dynamic
172.21.0.4      02bf.ac15.0005 vlan40  flood  dynamic
```

- No FDB entry is created for the MAC address in the content of the ARP packet. This ensures that packets destined to this MAC address are flooded to all ports of the VLAN.

This mode is configured by the command **arp-mac-disparity unicast** on the VLAN that faces the clustered servers.

For example:
```
con t
int vlan10
    arp-mac-disparity unicast
```
Multicast Mode

In multicast mode, not only is the ARP response from the clustered servers disparate (i.e. the source MAC on the Ethernet header differs from the sender MAC within the ARP packet content) it also has the unusual property that the sender IP address in the packet is unicast, but the sender MAC address is multicast.

By default, x-series switches simply drop such ARP responses, deeming a multicast sender MAC to be invalid.

However, the command `arp-mac-disparity multicast`, entered on the VLAN that faces the clustered servers, puts the switch into a mode whereby it will accept such ARP responses, and upon receiving them will do the same as it does when receiving unicast disparate ARPs when `arp-mac-disparity unicast` has been configured. That is, an ARP entry is created for the IP/MAC in the content of the ARP packet, but with the egress port set to ‘flood’, and no FDB entry is created for the MAC address.

As a result, packets destined to the IP address in question are flooded to the destination VLAN.

This behavior makes no distinction between clusters operating in Multicast mode and those operating in multicast mode with the IGMP option. The fact that a cluster is sending IGMP reports to the switch does not make any difference, it still floods the packets to all ports in the egress VLAN.

In both cases, when the ARP entries are learnt, they appear in the output of the command `show arp` as:

```
ATX8106M2#show arp
IP Address  MAC Address  Interface  Port  Type
10.100.0.50  03bf.0a64.0032  vlan200  flood  dynamic
10.100.0.56  0100.5e7f.0013  vlan200  flood  dynamic
10.100.0.75  03bf.0a64.0044  vlan200  flood  dynamic
10.100.0.76  03bf.0a64.0032  vlan200  flood  dynamic
10.100.0.60  0100.5e7f.0013  vlan200  flood  dynamic
```
Static multicast ARP entries

It is also possible to create static ARP entries with multicast destination MAC, and multiple egress ports, with the `arp` command:

```
arp UNICAST_IP MULTICAST_MAC PORT_LIST
```

For example:
```
arp 10.10.1.100 010e.11ff.2222 port1.0.1,port1.0.3,port1.0.8
```

In this case, an ARP entry is created for the IP/MAC in question, associated with all the egress ports specified in the command. At Layer 2 a special hardware forwarding entry is created, that associates the MAC address with the set of egress ports. So, packets destined to the IP/MAC in question are forwarded just to the egress ports specified in the command, and are not flooded to the whole egress VLAN.