Technical Guide

Allied Telesis

AlliedWare Plus™— Best Practice Guide

AlliedWare Plus™
OPERATING SYSTEM

CONTENTS

Introduction 4

Best Practices for Device Management 5

Remote CLI access 5

Management via the Web GUI 7

Management by SNMP 8

Best Practice Configurations to Aid Network Monitoring 9

Logging 9

Time synchronization 10

Protecting the CPU 13

Control plane bandwidth control 13

Using hardware ACLs to protect the CPU from attacks 14

Using hardware ACLs to protect the CPU from IPv4 attacks 14

Using hardware ACLs to protect the CPU from IPv6 attacks 15

Using Hardware ACLs to Block Packets from Invalid Source IP Addresses 15

General Good Practices for Switch Security 16

Close down and corral off unused ports 16

Avoid using VLAN 1 as the management VLAN for switches in the network 16

Use syslog to provide a detailed audit trail in the event of a suspected security breach or other problem 16

Use NTP to synchronize switch system time 16

Configure your switch to output a banner message when users connect 17

Avoid directed broadcast forwarding 17

Best Practices for Stacking 17

Use virtual MAC 18

Use the resiliency link 18

Avoid using VLAN IDs and IP subnets that are used internally within the stack 19

Turn off stacking on stand-alone unit 19

Class of Service (CoS) settings for VCStack operation 19

Best Practices for Spanning Tree 20

Introduction to Spanning Tree 20

Configure all edge ports as portfast 21

Protect against Root Bridge spoofing attacks 21

Use priorities to set a core switch to be root 21

Enable backups to Spanning Tree 22

Best Practices with Power over Ethernet 24

Introduction to Power over Ethernet (PoE) 24

Disable PoE on ports that should not be delivering power 25

Retain legacy interoperation 25

Configure power threshold alerts 25

Protect the power supply for important ports 25

Best Practices with IGMP 26

Introduction to IGMP 26

Ensure that there is a querier in every subnet where multicast is flowing 26

Do not disable IGMP snooping 27

Best Practices with IPv6 28

Introduction to IPv6 28

Configure explicit /64 IPv6 addresses on switches and routers 28

Configure RA Guard on non-routing switches 29

Best Practices for Routing Protocols 30

Introduction 30

Best practices for RIP configuration 30

Best Practices for OSPF Configuration 32

Configure OSPFv2 with MD5 authentication 32

Configure OSPFv3 with authentication and encryption 33

Unless the OSPF network is small, divide it into Areas 34

Configure route summarization at area boundaries 34

Set the reference bandwidth to an appropriate value for your link speeds 35

Introduction

Allied Telesis switches, with the AlliedWare Plus™ operating system, have a rich feature set.

When faced with an array of capabilities such as resiliency, security, management, switching, it can be difficult to know where to start. The purpose of this guide is to provide advice on getting the fundamental features of the switch configured in a secure, robust manner before you proceed on to the more specialised aspects of your switch configuration.

The topics covered in this guide are as follows:

- "Best Practices for Device Management" on page 5
- "Best Practice Configurations to Aid Network Monitoring" on page 9
- "Protecting the CPU" on page 13
- "General Good Practices for Switch Security" on page 16
- "Best Practices for Stacking" on page 17
- "Best Practices for Spanning Tree" on page 20
- "Best Practices with Power over Ethernet" on page 24
- "Best Practices with IGMP" on page 26
- "Best Practices with IPv6" on page 28
- "Best Practices for Routing Protocols" on page 30

Each topic includes a brief description of the function of the feature, followed by a walk-through of how best to create a foundation configuration for the feature.

Best Practices for Device Management

Being able to access one or more management interfaces is a basic requirement for successful deployment of any networking device. The management access needs to be not only reliable and flexible, but also secure. When installing networking equipment into an environment that exposes it to unauthorized access attempts, it is important that it is configured in a way that provides maximum protection against attacks.

In this section, we will work through the secure configuration of three distinct management interfaces into x-series products:

- Remote Command Line Interface (CLI) access
- The Graphical User Interface (GUI) accessed from a web browser
- SNMP management

Remote CLI access

AlliedWare Plus™ uses the Secure Shell (SSH) protocol for secure remote access to the CLI. SSH encrypts the traffic involved in the CLI connection using the strong RSA encryption algorithm, thereby thwarting attempts to snoop the conversation.

SSH servers identify themselves using a **host key**. Before the SSH client establishes a session with an SSH server, it confirms that the host key sent by the server matches its database entry for the server. If the database does not contain a host key for the server, then the SSH client requires you to confirm that the host key sent from the server is correct.

The AlliedWare Plus device will operate as the 'server' accept incoming SSH connections.

To configure SSH under AlliedWare Plus, start by creating the RSA host key for the device:

```
awplus(config)# crypto key generate hostkey rsa
```

Then, enable the SSH process. It will automatically discover the host key that you have created, and know to use that key for encrypting its sessions.

```
awplus(config)# service ssh
```

To check that the service is running, use the command: **show ssh server**

```
Secure Shell Server Configuration
-----------------------------------------------------------------
SSH Server                      : Enabled
Port                            : 22
Version                         : 2
Services                        : scp, sftp
User Authentication             : publickey, password
Idle Timeout                    : 60 seconds
Maximum Startups                : 10
Debug                           : NONE
```

Authenticating users connecting to SSH sessions

Users connecting to the switch by SSH can be authenticated in one of the following ways:

- checking their credentials in the local user database on the switch.
- sending their credentials to a RADIUS or TACACS+ server to be checked.
- using the RADIUS or TACACS+ server if available, or the local user database if the RADIUS or TACACS+ server does not respond.

By default, SSH users are checked against the local user database on the switch.

Creating an entry in the local user database is done by using the command:

```
awplus(config)# username <name> password <password>
```

But, having this entry in the local user database does not automatically enable that user to log into the switch by SSH. They need to be explicitly allowed SSH access:

```
awplus(config)# ssh server allow-users <username>
```

While setting up users in the local user database is simple, it quickly becomes tedious and inconvenient if the network contains a large number of network devices. In that case, it makes more sense to keep the user accounts in a centralised database on a RADIUS or TACACS server.

To set up the switch to send authentication requests to a RADIUS server, follow these steps:

1. Configure the switch with the address, pre-shared key (and optionally UDP port) of a RADIUS server.

   ```
   awplus(config)# radius-server host <ip-address> auth-port 1812 key <secret-key>
   ```

2. Configure an AAA authentication method list for login that uses the RADIUS server.

   There are two options:

   - Set the switch to use only the RADIUS server to authenticate all attempts to log into the switch's CLI:

     ```
     awplus(config)# aaa authentication login default group radius
     ```

     If the RADIUS server does not respond, then the login attempt fails.

   - Configure the switch to query the RADIUS server first, then use the local user database as a backup option if the RADIUS server does not respond:

     ```
     awplus(config)# aaa authentication login default group radius
     local
     ```

   If the RADIUS server rejects the login because it is not configured with a matching user, the login fails. If the RADIUS server is not available or does not recognise the switch as a RADIUS client, it will not respond, and the switch will check the local user database. If you use the local user database as a backup, you must also add the user to the local user database, as described above.

   Similarly, the switch can be configured with the details of TACACS+ servers:

   ```
   awplus(config)# tacacs-server host {<host-name>|<ip-address>} [key [81]<key-
   string>]
   ```

   An AAA authentication login definition can specify the TACACS-server group as the authentication method:

   ```
   awplus(config)# aaa authentication login default group tacacs+
   ```

Telnet

Telnet access to the switch is enabled by default. We recommend that you disable Telnet, and instead use SSH for command line remote access.

```
awplus(config)# no service telnet
```

Management via the Web GUI

Web access to a switch running AlliedWare Plus is achieved as follows:

1. Initially, the web browser connects to the switch by HTTP.
2. Via this HTTP connection, the web browser downloads a Java applet.
3. The rest of the management session from the web browser to the switch is controlled by the Java applet. The Java applet uses a combination of SNMPv3 and remote CLI sessions to exchange information with the switch.

After the initial download of the Java applet, there is very little HTTP exchanged between the web browser and the switch. In particular, no user authentication, switch monitoring or configuration information is exchanged by HTTP. All that information is exchanged by SNMPv3 or remote CLI sessions. As a result, it is not important to encrypt the HTTP connection. However, AlliedWare Plus does have the option of connecting to the GUI via HTTPS.

Note: The SSL certificate used for HTTPS is normally self signed, and will result in a warning message from a standard web browser.

The SNMPv3 user via which the Java applet communicates with the switch is forced to use both authentication and encryption. There is no option to use a user that implements the **noauth** or **auth** levels of security; the user must implement the **priv** security level.

The SNMPv3 aspects of the web management session are inherently secure. To ensure that the remote CLI connection involved in the web management session is secure, you must enable SSH on the switch. If SSH is not enabled, the Java applet uses Telnet. But as soon as the SSH service is enabled on the switch, the Java applet will stop using Telnet and use SSH instead.

To setup the switch for secure access to the Web GUI, follow these steps:

1. Disable Telnet.

```
awplus(config)# no service telnet
```

2. The GUI uses SSHv1, so a host key for SSHv1 is required.

```
awplus(config)# crypto key generate hostkey rsa1
```

3. The SSH service also requires a host key for SSHv2.

   Configure the switch for secure web browser management.

```
awplus(config)# crypto key generate hostkey rsa
```

4. Register the user as an SSH client.

```
awplus(config)# ssh server allow-users <gui-user-name>
```

5. Enable the SSH service.

```
awplus(config)# service ssh
```

   A that point, you can be confident that all SNMP and CLI interactions between the GUI applet and the switch will be encrypted, and secure.

To ensure that the initial HTTP communication between the browser and the switch is also encrypted, simply connect to the switch by HTTPS. That is, access the switch using the URL:

```
HTTPS://<switch-IP-address>
```

Management by SNMP

SNMP is a standardised protocol for interacting with network-connected devices. Each individual piece of information exchanged by SNMP has a unique identifier.

A very large number of such identifiers, and their associated pieces of information, have been precisely defined by RFCs. Therefore, network managers can be confident that devices that are compliant to these RFCs will respond to SNMP messages in a predictable manner.

In addition, the protocol has the flexibility to allow vendors to define their own vendor-specific identifiers, to enable the management of their own unique features.

SNMP is therefore a convenient protocol via which graphical management applications can interact with network nodes – sending instructions to the nodes, and collecting statistics from the nodes.

The pieces of information that can be obtained from the networking device by SNMP are grouped into sets called Management Information Bases (MIBs). Each MIB comprises a related set of information elements that pertain to a specific feature. Each information element has a unique identifier, called an Object ID (OID). These OID values, and descriptions of the piece of information referenced by each OID, are published in structured, human-readable and machine-readable text form called MIB files.

As described above, the AlliedWare Plus GUI makes use of SNMP, as do most Network Management Station applications. It is recommended to configure SNMP on all network devices, so that they can be available to network management stations.

As with all device management processes, it is highly advisable to ensure that SNMP is set up to operate in a secure manner. In the case of SNMP, that means using SNMPv3, as this version of the protocol introduces security features that were not present in previous versions of SNMP.

In particular, with SNMPv3, you can:

- encrypt the SNMP messages being sent across the network.
- check that SNMP messages are not tampered with during transit across the network.
- set up restricted views, that is, limited sets of MIB variables (OIDs) that can be accessed by particular users. These users need to enter a password to get access to their view.

Configuring SNMPv3

The following show a typical SNMPv3 configuration:

Note: By default, the switch does not respond to SNMP messages. So, to use SNMP, you do need to create an SNMP configuration on the switch.

In this example the user is named secure-user, belonging to a group named secure, the authentication algorithm is SHA (Secure Hash Algorithm) and the encryption algorithm is DES:

```
awplus(config)# snmp-server host 172.28.76.128 version 3 priv secure-user
awplus(config)# snmp-server group secure priv
awplus(config)# snmp-server user secure-user secure auth sha <hash-password> priv des
<encrypt-password>
```

Note that in the **snmp-server group** and **snmp-server host** commands, the security level can have three possible settings:

1. **noauth**—performs no authentication and no encryption
2. **auth**—performs authentication, but no encryption
3. **priv**—performs authentication and encryption, which provides the best security, and so is the recommended option.

With the configuration above, the user 'secure-user' can carry out SNMP communication with the switch from the management station at 172.28.76.128, provided they provide the correct hash password and encryption password.

SNMP notifications (Traps)

A valuable aspect of SNMP is its facility for devices to send out notifications of certain events such as low memory, port status changes, high temperature, authentication failures, and so on. These notifications are commonly known as traps.

We recommend making use of the SNMP trap capability, as it is a convenient way to ensure that events in the network are brought to the attention of network administration staff.

To enable SNMP to send traps for a range of features, use the command:

```
awplus(config)# snmp-server enable trap auth lldp loopprot mstp nsm rmon vcs vrrp
```

Some notifications must also be enabled within their protocols. For example, Link Layer Discovery Protocol (LLDP) notifications are enabled using the commands **lldp notifications** and/or **lldp med-notifications** as well as the command above.

Best Practice Configurations to Aid Network Monitoring

Logging

Allied Telesis Network devices maintain a detailed log of events that occur within the device. The types of events that are logged range from the receipt of invalid packets, to the timing out of protocol connections, to failures within software modules.

This information is extremely valuable for:

- monitoring the health of the network
- providing an audit trail for security breaches
- enabling the tracking down of problems occurring in the network

The log messages are stored within the switches themselves, and can also be sent to a central Syslog Server.

There is only limited memory on the switch itself for storing logs, so it is highly advisable to log all activity on the switch to a syslog server, so that a long history of log messages will be stored.

Logging severity levels

RFC5424 defines eight different levels of severity that can be assigned to log messages.

In order, from most severe to least severe, these are:

- Emergency
- Alert
- Critical
- Error
- Warning
- Notice
- Informational
- Debug

By default, the switch will store locally all log messages it generates that have a severity of **Notice** or higher. It is recommended to stay with this default setting unless, for some reason, the switch generates an excessive number of **Notice** severity or **Warning** severity messages.

When configuring the device to send log messages to a syslog server, it should be configured to send ALL severities of log message to the syslog server.

Configuration

The configuration command to inform the device to send log messages to a syslog server is:

```
log host <syslog-server-IP-address>
```

To configure the device to send ALL severities of log message (i.e. debug-severity and above) to the syslog server, use the command:

```
log host <syslog-server-IP-address> level debug
```

If you need to reduce the range of severity levels of log messages that are stored locally, then configure the highest-severity level that you want stored.

For example, to store only messages with severity level **Error** and above, use the command:

```
log buffered level Error
```

Time synchronization

Having the time synchronized on all the devices in the network is very useful when tracking down problems. When correlating events on one network node with events on another network node, it is essential that you are confident that the event timestamps on the two nodes are in sync.

Fortunately, there is a protocol that enables the synchronization of time between all the devices in a network. It is called Network Time Protocol (NTP). NTP can operate in a peer-to-peer mode, or in a client-server mode. In either case, the effect of NTP is that all the participating devices converge onto the same time, and keep their clocks in sync for as long as they continue to exchange NTP messages with each other.

Configuration

For peer-to-peer mode, the peer devices are simply configured with the IP addresses of the devices with which they will form a peer relationship:

```
awplus# configure terminal
awplus(config)# ntp peer 2001:0db8:010d::1
awplus(config)# ntp peer 172.16.7.3
```

For client-server mode, there is separate configuration for the clients and the server.

To set up a device as an NTP server, enter the commands:

```
awplus# configure terminal
awplus(config)# ntp master <Stratum>
```

Where "Stratum" is a number that designates where the server sits within the NTP hierarchy. If the device is acting as a standalone master (i.e. is not synchronizing to an external time source) then the value specified for Stratum is not important.

On the client devices, simply specify the IP address of the server to which they should synchronize:

```
awplus# configure terminal
awplus(config)# ntp server 192.0.1.23
```

OR

```
awplus# configure terminal
awplus(config)# ntp server 2001:0db8:010e::2
```

Securing NTP

NTP, like any service on the switch, is vulnerable to attack. The simplest attack is a DoS (Denial of Service) attack—flooding the NTP module with requests or updates. A more sophisticated attack involves spoofing a time source, thereby providing an incorrect time to the switch when the switch is acting as an NTP client.

AlliedWare Plus provides two mechanisms for guarding against NTP attack: NTP filtering and NTP authentication.

NTP filtering

Filtering makes use of access lists to specify the IP addresses with which the switch's NTP process will interact. A number of different types of access list can be applied to the NTP module, to control the different types of relationship that can occur in NTP. Access lists can include the following types:

peer

Time requests and NTP control queries will be accepted from devices whose addresses pass this access list. The switch's NTP process is not able to synchronize itself to a device whose address is not permitted by this access list.

This is configured with the command:

```
ntp access-group peer <ACL-number>
```

where `<ACL-number>` is the ID number of a standard IP ACL, created with the command:

```
awplus(config)# access-list {<1-99>|<1300-1999>} {deny|permit} <source>
```

query-only

NTP control queries are accepted from devices whose addresses pass this access list.

This is configured with the command:

```
ntp access-group query-only <ACL-number>
```

serve

Time requests and NTP control queries will be accepted from devices whose addresses pass this access list. The switch's NTP process is not able to synchronize itself to a device whose address is not permitted by this access list.

This is configured with the command:

```
ntp access-group serve <ACL-number>
```

serve-only

Only time requests are accepted from devices whose addresses are permitted by this access list. The access lists are applied using the command:

```
ntp access-group serve-only <ACL-number>
```

NTP authentication

The purpose of NTP authentication is to enable the client to authenticate the server, and not vice versa. NTP authentication specifically deals with malicious users attempting to spoof a valid NTP server. The authentication is performed by using an MD5 key. The server and the client must be both configured to perform authentication and to use the same MD5 key.

Configuring authentication on the NTP client

On the client, there are four commands required to configure authentication:

1. Enable authentication for NTP:

   ```
   awplus(config)# ntp authenticate
   ```

2. Create one or more MD5 keys that can be used for NTP authentication:

   ```
   awplus(config)# ntp authentication-key <keynumber> md5 <key>
   ```

   where the <keynumber> is an ID number that will be used in other commands to refer to this key. The <key> is just a string, for example: AB123434, and is limited to the maximum command line length, which is 464 characters without spaces (less the length of the preceding parameters).

3. Create a list of which of these keys is currently trusted (i.e. can currently be used for authentication):

   ```
   awplus(config)# ntp trusted-key <keynumber>
   ```

   where the <keynumber> is the ID number of an MD5 key that has been created for NTP authentication.

4. When defining the NTP server that the switch wishes to receive time updates from, specify the key that will be used to authenticate the session to this server:

   ```
   awplus(config)# ntp server <serveraddress> key <keynumber>
   ```

   where the <keynumber> is the ID number of an MD5 key that has been created for NTP authentication, and is in the list of trusted keys. The server must also be configured to use this key, as described in the following section.

Configuring authentication on the NTP server

When configuring the switch as an NTP server, the configuration required to enable authentication on the server is:

1. Enable authentication for NTP.

   ```
   awplus(config)# ntp authenticate
   ```

2. Create an MD5 key that can be used for NTP authentication.

   ```
   awplus(config)# ntp authentication-key <keynumber> md5 <key>
   ```

   where the <keynumber> is an ID number that will be used in other commands to refer to this key. The <key> is just a string.

3. Designate this key as currently trusted (i.e. can currently be used for authentication).

   ```
   awplus(config)# ntp trusted-key <keynumber>
   ```

   where the <keynumber> is the ID number of the MD5 key.

   All clients that wish to carry out authenticated sessions with this server must specify this key as the key they will use for sessions to this server.
   Note: Authentication is initiated by the client. If the server is configured for authentication but the client is not, then the server will still accept the client's session and serve time updates to the client. But if the client is configured for authentication and the server is not, then the session will never become established.

Configuring authentication for NTP peers

When using NTP in peer-to-peer mode, it is also possible for a pair of NTP peers to authenticate each other. In this case, the configuration is the same as in the NTP client case, except that the final command is replaced by one that defines a peer relationship.

```
awplus(config)# ntp peer <peeraddress> key <keynumber>
```

Of course, both peers must use the same key in the session with each other.

Filtering and authentication can be used together to provide a secure configuration. Filtering limits the addresses from which NTP messages will be accepted, and authentication enables a client to determine that a server is who it says it is.

Protecting the CPU

Control plane bandwidth control

In a Layer 2 or Layer 3 Ethernet switch, the vast bulk of packet forwarding is performed by the switching ASIC. These specialized switching chips are specifically designed to forward data of all packets sizes through all ports at or near line rate all the time.

As a result, the CPU of an Ethernet switch is not heavily involved in the hard work of packet forwarding. Rather, the CPU is involved in less intensive processes like device management, table updates, operating routing and resiliency protocols, system health monitoring, and so on.

Although the tasks in which a switch CPU is involved in are not intensive, they are vital to the operation of the switch. So, if these processes are interrupted, there are significant consequences.

To ensure that the CPU processing capability will never be oversubscribed by the data arriving from the switching fabric, a strict limit can be imposed on the rate at which data is transmitted from the Fabric-to-CPU channel. This works because network management and control traffic, whilst vital, is not high in volume. If high volumes of data are coming up to the switch's CPU, most of this data will not be valid control plane packets.

For instance, they could be packets generated by deliberate DoS attacks, or sustained high levels of broadcasts caused by a loop or a faulty device on the network. Limiting the rate of data transfer to the CPU will not penalize normal control plane communications, but will combat the effect of DoS attacks and storms.

By default, the AlliedWare Plus operating system sets the maximum rate to a level that is appropriate for each product. The chosen default rates are ample to accommodate the requirements for control plane traffic in any normal network environment, and yet low enough to prevent oversubscription of CPU-based processes.

We recommend that you avoid altering the limit unless there is no other option. However, if you need to alter this limit, use the command:

```
awplus(config)# platform control-plane-prioritization rate<rate-limit>
```

If the control plane data rate requirements in the network exceed the default rate, then you may need to look at which elements in the network design require such a high rate of control plane traffic, and consider other design options.

Using hardware ACLs to protect the CPU from attacks

Malicious attacks sometimes try to overload the CPU with traffic destined for the switch. Getting the CPU to run at its maximum capacity (100%) causes problems processing network control traffic, which is critical to keep a network functioning well. Control plane bandwidth limiting reduces the effect of such attacks. The effect can be further reduced by filtering out traffic that does not need to be sent to the CPU.

Using hardware Access Control Lists (ACLs) to block traffic destined for the switch's own IP address can protect the CPU. The following example shows how to configure ACLs to match on traffic destined for the switch's IP address. The ACLs allow SNMP, SSH, and web traffic to the switch's management IP address, but block all other traffic to the management IP address 172.28.78.23.

Using hardware ACLs to protect the CPU from IPv4 attacks

1. Permit SNMP traffic from the management subnet to the management address with UDP destination port 161:

   ```
   awplus(config)# access-list hardware protect-management
   awplus(config-ip-hw-acl)# 10 permit udp 172.28.0.0/16 172.28.78.23/32 eq 161
   ```

2. Permit HTTP traffic from the management subnet to the management address with TCP destination port 80:

   ```
   awplus(config-ip-hw-acl)# 20 permit tcp 172.28.0.0/16 172.28.78.23/32 eq 80
   ```

3. Permit SSH traffic from the management subnet to the management address with TCP destination port 22:

   ```
   awplus(config-ip-hw-acl)# 30 permit tcp 172.28.0.0/16 172.28.78.23/32 eq 22
   ```

4. If you wish to allow network managers to ping the switch, then permit ICMP traffic from the management subnet to the management address:

   ```
   awplus(config-ip-hw-acl)# 40 permit icmp 172.28.0.0/16 172.28.78.23/32
   ```

5. Create an ACL to block all other traffic destined to the management IP address:

   ```
   awplus(config-ip-hw-acl)# 50 deny ip any 172.28.78.23/32
   ```

   If there are some ports via which you wish to allow management access to the switch, then configure the hardware ACL on those ports:

   ```
   awplus(config)# interface port1.0.1-1.0.10
   awplus(config-if)# switchport mode access
   awplus(config-if)# access-group protect-management
   ```

If there are some ports via which you wish to block management access to the switch, then configure a blocking ACL on those ports:

```
awplus(config)# access-list hardware block-management
awplus(config-ip-hw-acl)# 10 deny ip any 172.28.78.23/32
awplus(config-ip-hw-acl)# interface port1.0.11-1.0.24
awplus(config-if)# switchport mode access
awplus(config-if)# access-group block-management
```

Using hardware ACLs to protect the CPU from IPv6 attacks

When SSH and SNMP are enabled on the switch, they are automatically accessible by both IPv4 and IPv6. If you wish to allow SSH and SNMP management of the switch by IPv6, but block all other IPv6 access to the management IPv6 address of the switch, you could proceed as follows (the management VLAN is VLAN2):

```
awplus(config)# ipv6 access-list snmpv6
awplus(config-ipv6-acl)# permit udp any 3ffe:89:90::1/128 eq 161 vlan 2

awplus(config)# ipv6 access-list sshv6
awplus(config-ipv6-acl)# permit tcp any 3ffe:89:90::1/128 eq 22 vlan 2

awplus(config)# ipv6 access-list other
awplus(config-ipv6-acl)# deny ip any 3ffe:89:90::1/128
awplus(config-ipv6-acl)# exit

awplus(config)# ipv6 traffic-filter snmpv6
awplus(config)# ipv6 traffic-filter sshv6
awplus(config)# ipv6 traffic-filter other
```

Using Hardware ACLs to Block Packets from Invalid Source IP Addresses

There is no good reason to accept packets from invalid source IPs, and there is a good chance of such packets being part of an attack. It is good practice to filter out such packets.

```
awplus(config)# access-list 3200 deny ip 127.0.0.0/8 any
awplus(config)# access-list 3201 deny ip 0.0.0.0/32 any
awplus(config)# access-list 3202 deny ip 224.0.0.0/3 any
awplus(config)# access-group 3200
awplus(config)# access-group 3201
awplus(config)# access-group 3202
```

Note: It is not necessary to explicitly block packets from multicast source MAC addresses, as these are blocked by default.

General Good Practices for Switch Security

The following are a set of quite simple configuration steps you can perform that will further reduce the risk of security violations in your network.

Close down and corral off unused ports

If there are any ports on the switch that are not yet in use, then make sure nobody can use them to access the network.

There are two good practices that can be performed in this regard:

- Shut these ports down.

```
awplus(config)# interface port1.0.14
awplus(config-if)# shutdown
```

- Put the ports into a dead-end VLAN, so that even if they are mistakenly enabled, the traffic that enters the ports cannot get any further than the local switch.

  To do this, create a Private Edge VLAN that has no IP address and no promiscuous port. Make this VLAN the native VLAN on the unused ports. That way, traffic entering the ports has no way to be forwarded on by the switch at either Layer 2 or Layer 3.

```
awplus(config)# vlan database
awplus(config-vlan)# private-vlan 3 isolated
awplus(config)# interface port1.0.2
awplus(config-if)# switchport access vlan 3
awplus(config-if)# switchport mode private-vlan host
```

Avoid using VLAN 1 as the management VLAN for switches in the network

VLAN 1 is the default VLAN on Allied Telesis, and other vendors' switches—it is the VLAN that is the most likely to be the native VLAN on any switch port that has not been configured with security in mind, so it is the VLAN that an attacker is most likely to be able to get their traffic into.

You can instead create another specific VLAN for device management, minimize the number of ports in the network that are members of this VLAN, and isolate this VLAN from the VLANs containing user traffic. This reduces the chances of an attacker being able to get management access to the switches.

Use syslog to provide a detailed audit trail in the event of a suspected security breach or other problem

Configure the switch to log all activity to a syslog server:

```
awplus# configure terminal
awplus(config)# log host <syslog-server-IP-address> level informational
```

Use NTP to synchronize switch system time

Investigating any events that happen on the network is easier if the system time on all switches is synchronized. The most effective way to synchronize the time on all the switches is to use NTP. A possible configuration to securely synchronize the switch to a timer server would be:

```
awplus(config)# ntp authenticate
awplus(config)# ntp authentication-key 23 md5 secretKey
awplus(config)# ntp trusted-key 23
awplus(config)# ntp server <server-IP-address> key 23
```

Configure your switch to output a banner message when users connect

This banner should give notice to anyone who connects to a switch that it is for authorized use only and any use of it will be monitored. Courts have dismissed cases against those who have attacked systems without banners. Having no banner on a switch may lead to legal or liability problems.

```
awplus# configure terminal
awplus(config)# banner login
```

<Type in the text for the banner>

Type CNTL/D to finish.

```
awplus(config)# exit
awplus# exit
```

Avoid directed broadcast forwarding

If possible, do not enable directed broadcast forwarding. Directed subnet broadcasts are a technique commonly used by DoS attacks (and for spreading viruses). A directed subnet broadcast occurs when a host sends a packet to the broadcast address of a subnet that is not the originating host's own subnet.

For example, the host 192.168.2.45 sending a broadcast to 192.168.3.255, to attack all devices in the 192.168.3.0 subnet.

For the message to be broadcast, the router that provides the gateway from the 192.168.2.0/24 subnet to the 192.168.3.0/24 subnet must turn the packet from a unicast to a broadcast and forward it onto the 192.168.3.0/24 subnet. This act on the part of the gateway router is commonly referred to as directed broadcast forwarding.

The switches are capable of performing directed broadcast forwarding, but by default that capability is disabled. It should only be enabled if it is strictly required.

Best Practices for Stacking

A range of Allied Telesis products running AlliedWare Plus support Virtual Chassis Stacking - VCStack™. This is a feature whereby two (or more) switches can operate together as a single unified entity. The prime benefits of VCStack are:

Resiliency – the stacked switches form an active-active switching cluster. If one stack member fails, then other stack members will continue forwarding traffic. Provided the devices attached to the stack all have connections to more than one stack member, then full connectivity persists even when a stack member is out of action.

Port density – a virtual chassis stack effectively becomes a unified device, containing a large number of ports, that operates as single switch and is managed as a single switch. This enables a high-port-density switch to be created in a modular 'pay as you grow' fashion.

Simplified configuration – because the switches within a virtual chassis stack automatically synchronize their forwarding tables, and automatically prevent any packet loops forming between the stack members, there is a reduction in the amount of routing protocol and resiliency protocol configuration required in the network.

Simplified network maintenance – if a switch within a stack fails, then the replacement of that failed switch is a simple matter. The failed switch is removed, and a fresh switch is connected into the stack to replace it. The stack will automatically install the correct configuration into the replacement switch, and can even upgrade the switch to the same software version as the rest of the stack members.

There are a number of recommended best practices in the configuration of **VCStack**.

Use virtual MAC

As part of a virtual chassis stacking network design, the VCStack uses a virtual MAC address for communication with other devices. As this single virtual MAC address is used for the complete VCStack, there is no change of MAC address if a new stack member is required to become master. In conjunction with the rapidity of the VCStack master failover, this ensures maximum continuity of network service, as there is no need for other devices in the network to learn a new MAC address into their MAC or ARP tables.

The virtual MAC address can be manually configured by specifying a VCStack virtual chassis ID. The ID selected will determine which virtual MAC address the stack will use. The MAC address assigned to a stack must be unique within its network.

The virtual chassis ID entered will form the last 12 bits of a pre-selected MAC prefix component; that is, 0000.cd370xxx

For example:

```
awplus(config)# stack virtual-mac
awplus(config)# stack virtual-chassis-id 63
```

This will result in a virtual MAC address of: 0000.cd37.003f

Use the resiliency link

If one or more stacking links fail, so that some stack members are no longer in contact with the active master switch, then the other stack members are left with a dilemma. Should they assume that the active master switch has gone down, and re-elect a new active master, or should they assume that the active master is still operating?

This problem is particularly important when a stack has multiple connections to edge switches in the network. In this network scenario, if a stack splits into two active sections that are operating independently, then the edge switches will continue to treat their uplink ports as aggregated and share traffic across the links. However, the broken stack link could mean that traffic arriving at some of the stack members cannot reach the intended destination. It is necessary to have a backup mechanism through which stack members can check if the active master is still active, incase the stack link communication to the active master is lost.

The mechanism provided in Allied Telesis VCStack is called the resiliency link, which may be either the eth0 port (only on SwitchBlade, x930, or x900 series switches) or a dedicated VLAN (resiliencylink VLAN) to which switch ports may become members. These resiliency ports are connected together, and the stack members all listen for periodic (one per second) Health Check messages from the master. As long as the active master sends Health Check messages, the other stack members know that the active master is still active.

This means that the stack members can know whether the active master is still operating if they lose contact with the active master through the stacking links.

The out-of-band Ethernet port is configured as a resiliency port with the command:

```
awplus(config)#stack resiliencylink eth0
```

Note that even if you configure the eth0 port as a resiliency port, you can still use it for out-of-band management.

A VLAN, and switch port are configured for resiliency link connection with the commands:

```
awplus(config)#stack resiliencylink vlan1000
awplus(config)#interface port1.0.1
awplus(config-if)#switchport resiliencylink
```

Note, also that this VLAN is dedicated to the resiliency link function and must not be the stack management VLAN or a customer data VLAN.

In the situation where a stack member loses contact through the stacking links, but continues to receive health check messages via the resiliency link, the stack member becomes a disabled master. It runs the same configuration as the active master, but it has its data ports shutdown.

Avoid using VLAN IDs and IP subnets that are used internally within the stack

Internal communication between the stack members is carried out using IP packets sent over the stacking links. This stack management traffic is tagged with a specific ID and uses IP addresses in a specified subnet. By default, the VLAN and subnet used are:
`VLAN - 4094, Subnet - 192.168.255.0/27`

This VLAN and subnet are also used for internal communication between cards in the SBx8100 chassis. Therefore, if using stacked switches, or an SBx8100 chassis, it is recommended to avoid using this VLAN and subnet in the network. You may need to change these values if they clash with a VLAN ID or subnet that is already in use in the network. Use the commands:

```
awplus(config)# stack management subnet <ip-address>
awplus(config)# stack management vlan <2-4094>
```

It is important that the settings for management subnet and management VLAN are the same for all the switches in a stack. If a switch is added to a stack, and its setting for management VLAN and/or management subnet differ from those on the other stack members, the new switch will not be joined to the stack and a log message similar to the following will be created:

```
06:51:41 awplus-vcs VCS[1066]: Member 2 (0009.41fd.dd0b) cannot join stack - incom-
patible VCS management subnet
```

Turn off stacking on stand-alone unit

On a standalone switch, that is not intended to be a member of a stack, explicitly disable stacking using the command:

```
no stack enable
```

This will result in the switch booting up slightly more quickly, as it does not spend any time at startup trying to determine if it is connected to a stack.

Class of Service (CoS) settings for VCStack operation

In general you can apply the same principles when configuring QoS on a VCStack as you would for single switch; however there are a few specific changes that you will need to make. Switches within a VCStack, exchange their stack management information and user data over their high speed inter-stacking links. The stack management information is preassigned to the egress queue 7. This is the highest value queue, and (in a stacked configuration) its traffic should not be shared with any user data. However, any CoS tagging of 7 applied to the incoming data will automatically be assign to queue 7 as it crosses the internal stacking links. You will therefore need to reconfigure your CoS to Queue settings to ensure that no user data is sent to queue 7.

To prevent this from happening, we recommend that you make appropriate changes to your queue settings (mappings) to reflect the stacking requirement previously described.

This process should include (but not be limited to) running the following command to ensure that any remaining user still carrying a CoS 7 tag, will be mapped to egress queue 6. To remap priority CoS traffic to egress queue 6, run the following command.

```
awplus# config terminal
awplus(config)# mls qos map cos-queue 7 to 6
```

Best Practices for Spanning Tree

Introduction to Spanning Tree

Spanning Tree Protocol (STP) is a loop protection protocol. It allows switches in a multiply-connected, Layer 2 network to create a loop-free tree that provides a single unique path between any pair of network nodes. Using STP, switches can gather information about each other, negotiate which links to block, and perform ongoing monitoring of the state of the network. This allows you to create a network that has redundant backup paths that can automatically cut over to a backup path when the primary path fails.

STP was first standardized in 1990 by the IEEE. Over time, extensions to STP, initially developed as proprietary extensions, were also standardised by the IEEE: Rapid STP (RSTP) and Multiple STP (MSTP).

RSTP has faster convergence, less network disruption when topology changes occur, and is more tunable to network needs than STP. For example, the PortFast feature allows RSTP to treat any ports that are not connected to other switches, as edge ports, that go into a forwarding state as soon as they are UP.

Multiple Spanning Tree Protocol enables a set of spanning trees to effectively operate independently on the same set of physical network segments. Each of the almost independent spanning trees protects a different set of VLANs.

The following simple set of rules allows the spanning tree protocol to create a loop-free set of forwarding paths that do not leave any network node isolated:

1. One of the switches is elected as the Root Bridge for the spanning tree.

2. All ports on the Root Bridge go into the forwarding state.

3. All other switches then arrange the spanning tree topology based on their path cost to the Root Bridge.

   The ports are assigned specific roles:

   - the switches work out which of their ports has the 'least cost' path to the Root Bridge. These ports are called Root Ports, and are put into the forwarding state. In essence, Root Ports are ports heading towards the Root Bridge. Each switch has only one Root Port at any one time.
   - on direct point-to-point links between pairs of switches, if the port at one end of the link is a Root Port, then the port on the neighboring switch at the other end of the link is put into the forwarding state. These ports are given the role of Designated ports. In fact, Designated Ports are any port in the forwarding state that are not Root Ports.
   - on links where neither port at either end of the link is a Root Port, a negotiation occurs to work out which is the 'superior' port. This superior port is put into the forwarding state as a Designated Port. The port at the other end of the link is given the role of an alternate port–literally, an alternative to the switch's current Root Port–and is blocked from forwarding. Alternate ports may be quickly transitioned to forwarding if there is a problem with the Root Port connection to the spanning tree.
   - if multiple ports connect onto a single segment (all connect to a hub, for example), then if any of these ports is a switch's Root Port, it will be put into forwarding. Among the remaining ports there will be a negotiation to determine the 'superior' port, which will be put into forwarding.

Once all this negotiation is complete, then the protocol has done its job–all the loops have been removed from the network with the least number of blocked ports, and the network is said to have **converged**.

Configure all edge ports as portfast

Once a pair of switches are connected together, and the ports connecting them have gone link-up, they will stay continuously link-up for long periods of time. So, the link topology between switches very rarely changes state.

However, the switch ports at the edge of the network (where the workstations connect) are likely to change state more than once a day. Every time a workstation is turned on or off, or reboots, its link to the network switch will change state between link-up and link-down a few times. State changes on active spanning tree ports usually triggers the topology change process in the spanning tree, which result in switches flushing their forwarding database tables and then flooding packets as they relearn MAC addresses.

This process can disrupt the smooth running of the network, so it is beneficial to the network to stop the edge ports from operating as active spanning tree ports. As well, it is rare for network loops to exist beyond those edge ports as each port is normally connected to just one terminating device.

For these reasons, the PortFast feature exists in RSTP. This is a defined setting that is tailored specifically for the requirements of edge ports.

It puts the ports into a state where they:

- immediately go into the forwarding state as soon as they go link-up
- do not generate topology change events when they go into forwarding

These ports will still emit Hello packets, on the off-chance that a loop is inadvertently formed. If they receive Bridge Protocol Data Units (BPDUs), then depending on the port's configuration, they will either become active STP ports or they will shut down. So, the **port-fast** setting still provides protection against loops.

To enable the **portfast** setting on a port, enter the port's configuration mode and enter the command:

```
awplus(config-if)#spanning-tree portfast
```

To configure the port to shut down if it receives BPDUs, specify the **bpdu-guard** option:

```
awplus(config-if)#spanning-tree portfast bpdu-guard
```

If you do not use this option, then if the port receives a BPDU it will begin to negotiate spanning tree with the device sending BPDUs.

Protect against Root Bridge spoofing attacks

Root Bridge spoofing is a data-stealing and denial-of-service attack. The attacker sets up a device that emits STP Hello packets with a very low priority to enable their device to get elected as the Root Bridge. Once it is elected, the attacker will be able to see a lot of the data on the network, allowing them to steal that data. Root Bridge spoofing also disrupts the network. To protect against this attack, you can configure specific ports on each switch to shut down if they receive BPDUs with a lower priority than the switch's own priority. You should configure this setting only on ports that you know should never be Root Ports on the switch.

To enable this feature on a port, enter the port's configuration mode, and use the command:

```
awplus(config-if)#spanning-tree guard root
```

Use priorities to set a core switch to be root

It is more efficient for a core switch in the network to be the root of the spanning tree, rather than a switch at the edge of the network. When a spanning tree network has converged, all forwarding paths in the network emanate from the root switch. It is sensible for these paths to emanate from a high-performance switch at the centre of the network than from a low-performance switch at the edge of the network.

This will, on average, generate the shortest active paths between devices on the network, and will ensure that the overhead of being Root Bridge does not fall onto a low-performance device.

By default, the switch with the lowest MAC address in the network will become the Root Bridge. This switch could be **anywhere** in the network, it is really random chance as to which switch will have the lowest MAC address.

To force a core switch to become Root, you can use a feature that sets switches' root priority. The switch with the lowest value of priority becomes Root (and then MAC address is used as a tie breaker if multiple switches have equal lowest priority).

The recommended practice is to set the priority to zero on the switch that you wish to become the Root Bridge:

```
awplus(config)#spanning-tree priority 0
```

If there is another switch that you want to take over as Root if the zero-priority switch fails, then set that switch's priority to the next-lowest value – 4096 (spanning tree priorities increment in multiples of 4096).

```
awplus(config)#spanning-tree priority 4096
```

Enable backups to Spanning Tree

Spanning tree can fail in a network, due to misconfiguration, or processing failures or interop issues. If Spanning Tree does fail, and uncontrolled loops are formed in a network, the effect is serious. Typically, severe packet storms result, causing almost complete disruption to any user traffic on the network.

It is highly recommended that a backup for Spanning Tree is configured. AlliedWare Plus has three backups for Spanning Tree, which can all be used simultaneously:

- Broadcast/multicast/DLF limiting
- Loop protection
- MAC-thrash protection

Broadcast/multicast/DLF limiting

A storm results in broadcasts, multicasts, and flooded DLFs (destination lookup failures) being forwarded around a loop. The severity of the storm is reduced by putting limits on how many broadcast/multicast/DLF are forwarded per second. The maximum forwarding rates for broadcast, multicast, and DLF can be set on a per-port basis with the commands:

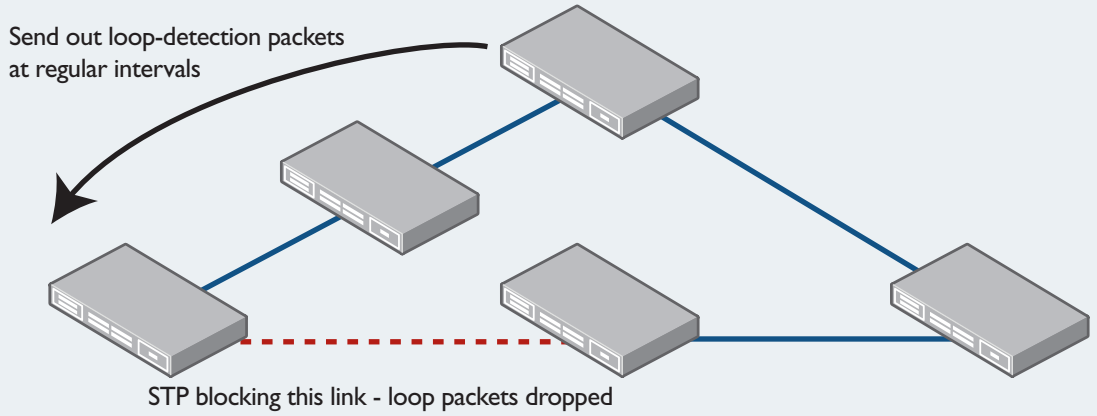
```
int portx.y.z
  storm-control broadcast level <level>
  storm-control multicast level <level>
  storm-control dlf level <level>
```

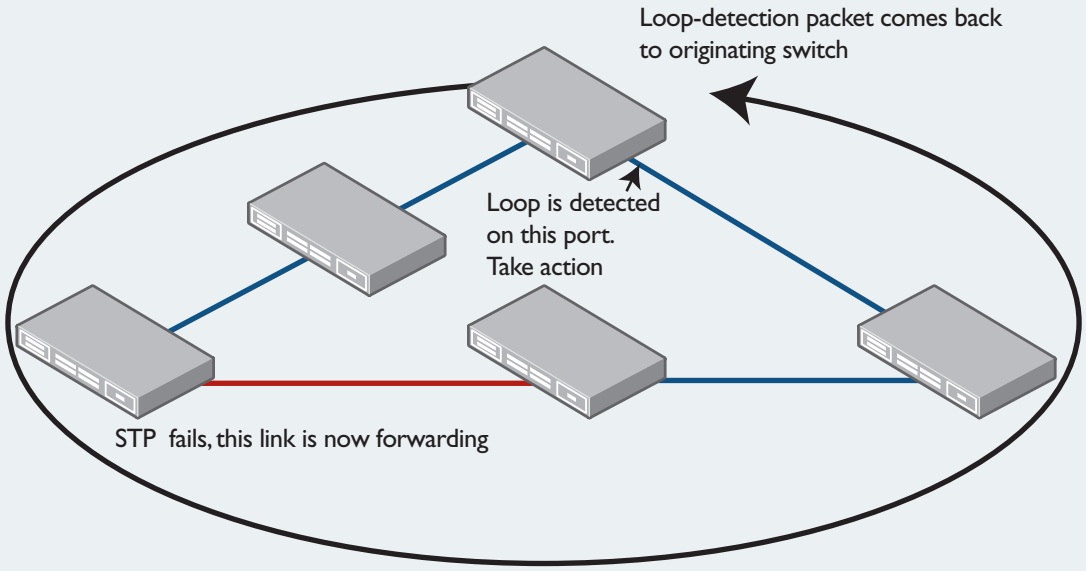
Where "level" is the maximum forwarding rate of the specified traffic type, as a percentage of port bandwidth.

Loop protection

To detect loops this feature operates by transmitting a series of Loop Detection Frames (LDFs) from each switch port out into the network. If no loops exist, then none of these frames should ever return. If a frame returns to its originating switch, the detection mechanism assumes that there is a loop somewhere in the network and offers a number of protective options.

Probe packets are sent out. Normally they are dropped somewhere in the loop.





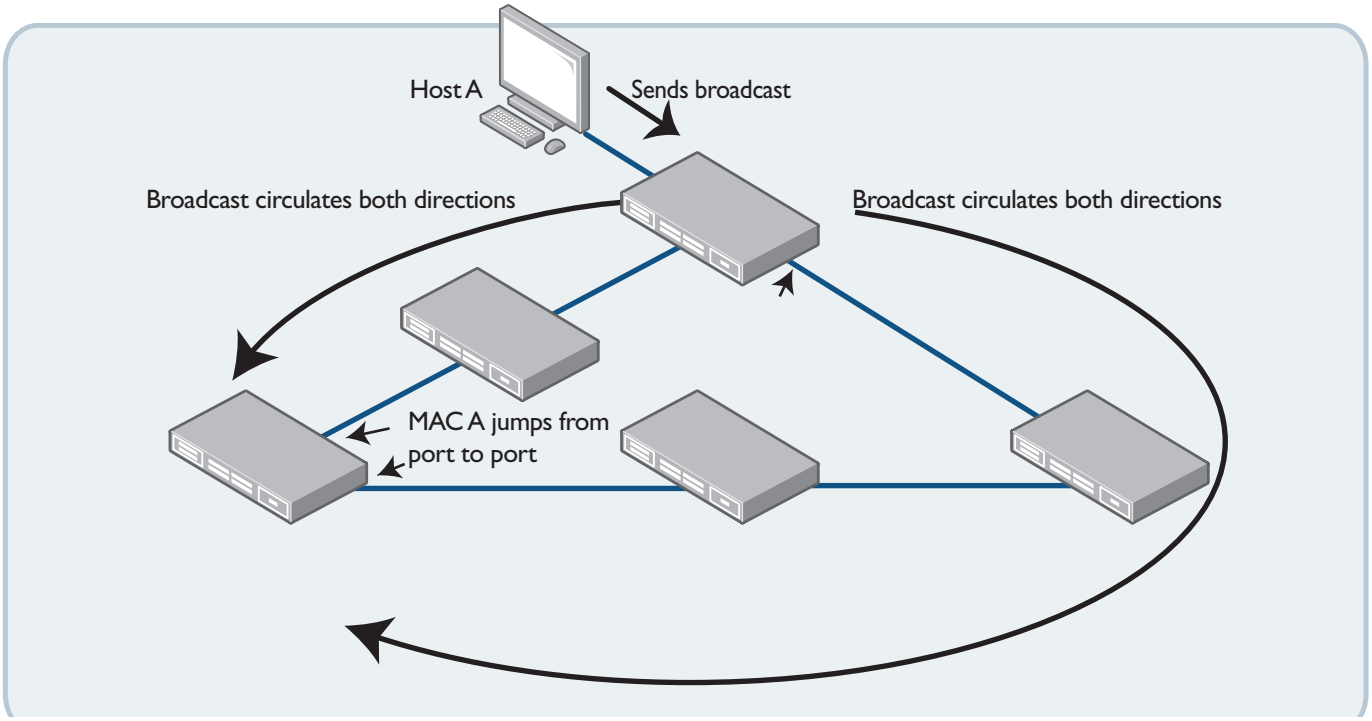
If the packets get right around the loop, there is a problem.

The recommended configuration is for loop-protection to disable a port for 30 seconds if it detects a loop:

```
int portx.y.z
loop-protection action port-disable
loop-protection timeout 30
```

MAC-thrash detection

MAC address thrashing occurs when MAC addresses move rapidly between two or more ports.



The recommended configuration is for thrash limiting to disable a port for 30 seconds if it detects a loop:

```
int portx.y.z
thrash-limiting action port-disable timeout 30
```

Best Practices with Power over Ethernet

Introduction to Power over Ethernet (PoE)

PoE is a mechanism for supplying power to network devices over the same cabling used to carry network traffic. PoE supplies power to network devices called Powered Devices (PDs). Allied Telesis x-series switches support two PoE standards, IEEE 802.3af and IEEE 802.3at.

- The 802.3af, PoE, standard specifies how power should be distributed to PDs over twisted pair Ethernet LAN cables. The IEEE 802.3af standard was approved in June 2003. This standard supports up to 15W at 48V DC.
- The IEEE 802.3at standard, Power over Ethernet Plus (PoE+), specifies how higher power levels (up to 60 W over a voltage range of 50 to 57 VDC) should be distributed over Ethernet LAN cables to networked devices. The IEEE 802.3at standard was approved in September 2009.

PD detection is carried out in real-time by the PSE controller on all switch ports to detect and monitor the presence of any powered devices. Power is not supplied to any specific port until a valid PD is detected. A switch port which has a PD unplugged will cease to have power supplied.

Disable PoE on ports that should not be delivering power

By default, on PoE-capable x-series switches, all RJ45 ports will deliver power when connected to a PD. If there are ports from which you definitely do not wish to deliver power, then explicitly disable PoE on those ports.

```
interface portx.y.z
awplus(config-if)# no power-inline enable
```

A port that has PoE disabled will operate as a normal Ethernet port and will not supply power to its cable connection.

Retain legacy interoperation

The AlliedWare Plus implementation of PoE offers two methods of PD detection. The default is to use the IEEE 802.3af and IEEE 802.3at standards' resistance and capacitance measurements. The second option is to support legacy PDs that were designed before the IEEE standard was finalized. This involves measuring for a large capacitance value to confirm the presence of a legacy PD. The IEEE method will be tried first, and failing the discovery of a valid PD, the legacy capacitance measurement will be tried.

By default, legacy PD detection is enabled on all ports. It is recommended to retain this setting, to avoid unexpected failure to power up devices that happen to conform to the pre-standard specification.

If a legacy mode has been disabled, it can be re-enabled with the command:

```
awplus(config)#  power-inline allow-legacy
```

Configure power threshold alerts

The switch can be configured to send a Simple Network Management Protocol (SNMP) trap to your management workstation and record an entry in the event log whenever the total power requirements of the powered devices exceeds a specified percentage of the total maximum power available on the switch. With the default setting of 80% applied, the switch sends an SNMP trap when the PoE devices require more than 80% of the maximum available power on the switch.

To adjust the threshold, use the command:

```
awplus(config)# power-inline usage-threshold <1-99>
```

For your management workstation to receive traps from your switches, you must configure SNMP traps (notifications) for PoE, using the command:

```
awplus(config)# snmp-server enable trap power-inline
```

Protect the power supply for important ports

Port prioritization is the way the switch determines which ports are to receive power in the event that the needs of the PDs exceed the available power resources of the switch.

If the PDs connected to a switch require more power than the switch is capable of delivering, the switch will deny power to some ports. You can use port prioritization to ensure that PDs critical to the operations of your network are given preferential treatment by the switch in the distribution of power, should the demands of the PDs exceed the available capacity.

It is recommended to configure critical priority (the highest priority) on those ports connected to equipment that must **never** lose power:

```
awplus(config)# interface portx.y.z
awplus(config-if)# power-inline priority critical
```

Best Practices with IGMP

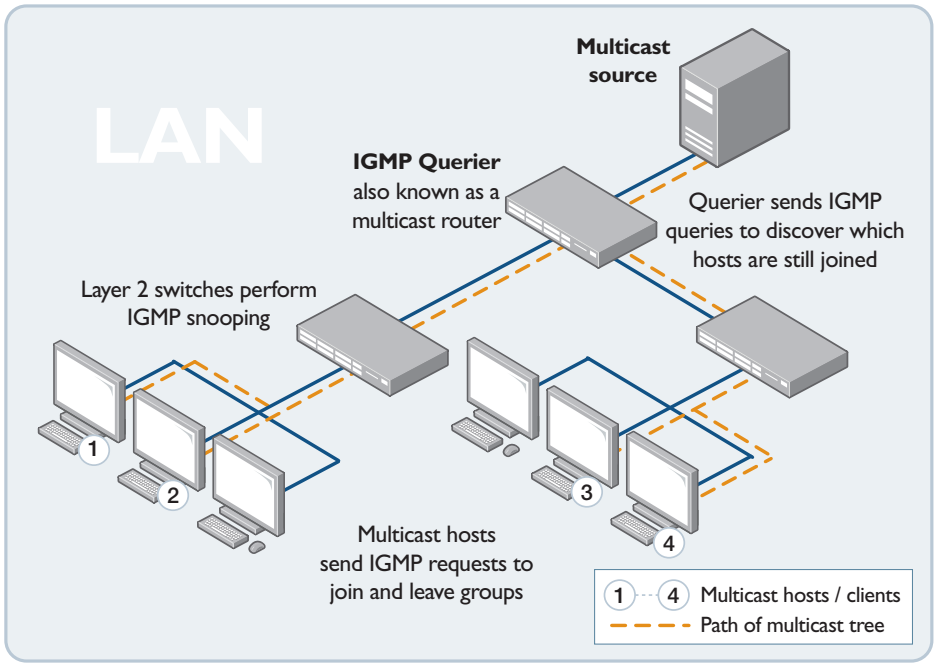
Introduction to IGMP

Multicasting provides an efficient way to transmit packets to a group of hosts simultaneously while conserving network bandwidth. Unlike unicasting, hosts do not directly communicate with the traffic source, but instead join a multicast group. The network devices between the hosts and the source keep note of which hosts are in a multicast group, and only forward the data for that group to these hosts.

Internet Group Management Protocol (IGMP) is the mechanism by which hosts can join a multicast group, and through which routers and Layer 3 switches keep track of where to forward multicast streams. Layer 2 networks can also snoop on IGMP messages so that they can continue to forward multicast streams efficiently.

Ensure that there is a querier in every subnet where multicast is flowing

In every Layer 2 multicast network, there needs to be a device that is sending IGMP queries into the network. This is essential to maintaining multicast flows once they have been established.



Typically, the device that is configured to send the queries is the router that is the gateway from the local network into a Layer 3 network. However, as in the self contained multicast network shown in the diagram, it does not necessarily need to be performing Layer 3 forwarding of multicast. The device configured to send the queries is sometimes referred to as the multicast router (or mrouter), and sometimes as the **Querier**.

Do not disable IGMP snooping

IGMP snooping is enabled by default on all Allied Telesis x-series switches.

Switches that are performing IGMP snooping will forward groups only to the ports on which they have recently received IGMP reports requesting packets for the group in question.

The intended operation of Layer 2 multicasting is that devices which wish to receive certain groups will send out IGMP reports requesting those groups. Some processes that communicate by multicast do not adhere to this intended mode of operation – they simply send out multicast data, and expect it to be received by the interested hosts, irrespective of whether those hosts are sending IGMP reports or not.

For example, multicast is sometimes used to learn, or to advertise to users, services available on their network through Directory Agents (DAs) or directly with Service Agents (SAs) using Service Location Protocol (SLP).

User Agents (UAs) and SAs send multicast requests packets (not IGMP reports) to 224.0.1.35 to locate DAs on the network. UAs and SAs learn of DAs via periodic multicast (239.255.255.253) advertisements. The Service Location Protocol does not specify that UAs, SAs, or DAs send IGMP reports to request each other's multicast notifications; it simply expects that the network will flood the notifications, and thereby deliver them to the listening agents.

In the case that this sort of service is running in your network, it is very tempting to treat IGMP snooping as an impediment, and just disable IGMP snooping, so that the multicast advertisements will be flooded to all hosts.

However, disabling IGMP snooping can have serious consequences. The most likely consequence is that it will interfere with the management of your switches. The effect of disabling IGMP snooping is that multicast packets are flooded to ALL nodes in the network, including the CPU of the switches on which snooping has been disabled. So, if there are large volumes of multicast in the network, then this multicast traffic will be flooded to the CPUs of the snooping-disabled switches, and cause management traffic destined to those switches to be dropped.

The better alternatives to disabling IGMP snooping are either:

1. If you know the particular multicast groups that are used by the services you want to enable, then configure static IGMP entries for these groups onto the ports where you want the multicast to be forwarded:

   ```
   awplus(config)# interface vlanx
   awplus(config-if)# ip igmp static-group a.b.c.d  interface portx.y.z
   ```

   OR

2. If you have some ports to which you want to forward all multicast groups, rather than specific groups, then configure these as **mrouter** ports:

   ```
   awplus(config)# interface vlanx
   awplus(config-if)# ip igmp snooping mrouter interface portx.y.z
   ```

Best Practices with IPv6

Introduction to IPv6

Internet Protocol version 6, or IPv6, is an improved version of the current and most widely used Internet Protocol, IPv4. IPv6 was also known as IPng (Next Generation).

Address depletion is the primary driver behind the need for IPv6. Commercial opportunities have rapidly increased the need for IP addresses with the demand for wireless devices, peer-to-peer networking and the 'smart home' which, because they access the Internet, require their own IP address. There are more devices connected to the Internet than IPv4 addresses, as NAT has allowed many addresses to 'hide' behind a single public address.

- IPv6 provides an enormous amount of extra address space over IPv4. From IPv4's 32 bits to IPv6's 128 bits, the number of available IP addresses increases from 4 billion to over 340 trillion trillion trillion.
- IPv6 also improves on IPv4 by adding enhancements for security, multimedia traffic management, and simplified network configuration. The transition from IPv4 to IPv6 will be gradual and both IPv4 and IPv6 will coexist for some period of time yet.

IPv6 addresses are 128 bits long whereas IPv4 addresses are only 32 bits long. The new 128-bit IPv6 addresses are written as eight hexadecimal groups. Each hexadecimal group is separated by a colon (:) and consists of a 16-bit hexadecimal value.

A complete IPv6 address could look like this:

```
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
```

A group of xxxx represents a 16-bit hexadecimal value with each individual x representing a 4-bit hexadecimal value. The following is an example of a possible IPv6 address:

```
2001:0340:0000:0000:0000:F673:0029:0564
```

To make IPv6 addresses easier to write, the leading zeros in a 4-digit block can be removed. Also, contiguous sets of 4 zeros, and their separating colons, can be completely removed, and replaced by : :.

Therefore, `2001:0340:0000:0000:0000:F673:0029:0564` can be written as `2001:340::F673:29:564`.

Configure explicit /64 IPv6 addresses on switches and routers

Although IPv6 provides a facility to autogenerate IPv6 addresses (a process known as Stateless Auto Address Configuration, or SLAAC), it is recommended that network infrastructure components like switches and routers are statically configured with addresses, so there is never any doubt about which IPv6 address belongs to which item of infrastructure.

Moreover, the standard recommendation in deployment of IPv6 addresses is to use /64 netmasks.

The commands to configure an IPv6 address on an interface of an AlliedWare Plus device are:

```
awplus#conf t
awplus(config)#int vlan1
awplus(config-if)#ipv6 address 2003:78:ab34:9e43::1/64
```

Enable Router Advertisements on routing devices

Whilst it is recommended for infrastructure items to have static IPv6 addresses, it is equally recommended for workstations to have auto-generated addresses, for simplicity of deployment. To enable workstations to auto-generate their IPv6 address, the router on a subnet must transmit IPv6 Router Advertisements.

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 indicates whether it is a default gateway.

When the host receives the Router Advertisement, it extracts the Network Address from the packet and combines this with its auto-generated host ID to create its full IPv6 address. Router Advertisements are configured on AlliedWare Plus on a per-interface basis.

To enable RA advertisements use the command:

```
awplus(config-if)#no ipv6 ndsuppress_ra
```

Configure RA Guard on non-routing switches

As described above, Router Advertisements (RA) are key to the automatic address configuration on IPv6 hosts. RA messages advertise a router's presence and specify network parameters that are used by hosts as part of address auto-configuration. Similarly, Redirects provide information on the next hop routers for particular destinations.

Subverting the RA and redirect processes 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 implemented on switches that sit between routers and hosts. RA Guard drops bad RAs and redirects 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. It is recommended to configure RA Guard on any port that is known to not be connected to a valid router.

RA Guard is enabled on an interface as follows:

```
awplus#conf t
awplus(config)#int port1.0.2
awplus(config-if)#ipv6 nd raguard
```

Best Practices for Routing Protocols

Introduction

Routing protocols enable data networks to be self-organising systems. Routers are aware of the blocks of addresses that are directly connected to them and, with the aid of routing protocols, advertise this knowledge to other routers. The recipients of these advertisements relay the information on to yet other routers, and so on, until the knowledge of which address blocks are located where is spread through the whole network.

The two most popular routing protocols in enterprise networks are:

- RIP (and the IPv6 version RIPng), which is a simple protocol in which routers just advertise their route table contents to each other.
- OSPF, which is a more complex protocol in which routers learn the full topology of their network, and calculate the best path to any point within the network. Although OSPF is more complex than RIP, it is also much more efficient and scalable than RIP.

Best practices for RIP configuration

This simplicity of RIP makes it quite attractive for small networks with a few routes, where highly responsive recovery from lost links or nodes is not essential. It is simple to understand, simple to configure, and has very little that can go wrong.

However, for larger networks, RIP does not really "cut the mustard". The CPU overhead of advertising the whole of a large route table at regular intervals can noticeably impair the performance of routers. The slow propagation of new routes through a network is not acceptable in a dynamic environment. The simple metric calculation takes little account of path costs. RIP is not the routing protocol of choice in large and/or high performance networks. But it does have its place in small, simple networks.

Use RIP version 2, not RIP version 1

The version of RIP protocol that an interface sends and receives can be configured on a per-interface basis. By default, an AlliedWare Plus devices sends and receives RIP version 2.

Because RIP version 1 has significant limitations, it is highly recommended to leave the RIP version configuration at the default setting, so the RIP version 2 will be used, not RIP version 1.

Reduce the RIP update timer

Setting the timer low does not necessarily propagate routing changes through the network more quickly, as changes should be propagated by triggered updates. However, increasing the frequency of updates will reduce the time taken for discrepancies between routers' route tables to be sorted out. Increasing the frequency of the updates does add to the amount of RIP traffic on the network, and add more load to routers' CPUs.

However, given that RIP should only be used in small networks (those with small route tables to pass around), then the extra load caused by a higher update frequency should not be a major problem. On balance, the advantages of increasing the update frequency outweigh the disadvantages. While, for many protocols, it is advisable to stay with default values for timers, it should be borne in mind that RIP was defined many years ago, when link bandwidths and CPU speeds were much lower than they are today. Therefore, lowering the update time to 5 or 10 seconds is a good idea.

The commands for altering the RIP timers are as follows:

```
awplus# configure terminal
awplus(config)# router rip
awplus(config-router)# timers basic <update> <timeout> <garbage>
```

The three values all need to be provided when the command is entered. If you want to leave one or two of the timers at default values, then provide those default values when entering the command.

For example. to set the update timer to 10 seconds, proceed as follows:

```
awplus# configure terminal
awplus(config)# router rip
awplus(config-router)# timers basic 10 180 120
```

Enable RIP authentication

Using RIP authentication guards against route table corruption that might be caused by devices on the LAN mistakenly or maliciously sending out RIP updates containing invalid routing information.

The steps to configure RIP authentication are as follows:

1. Configure the switch to send RIP out the interfaces with IP addresses within specified IP subnets.

   ```
   awplus(config)# router rip
   awplus(config-router)# network 172.28.0.0/16
   awplus(config-router)# network 221.189.62.0/24
   ```

2. Create a key chain. In this example, the name of the key chain is rip-key-chain. A key chain consists of a set of keys. Associated with each key are:

   - a send-lifetime—the period during which the key will be sent
   - an accept-lifetime—the period during which the key is a valid match.

   The accept-lifetime of one key can overlap the send-lifetime of another key, to allow for slight non-synchronisation between the times at which neighboring switches cut over from sending one key to sending another one.

   ```
   awplus(config)# key chain rip-key-chain
   awplus(config-keychain)# key 1
   awplus(config-keychain-key)# key-string piano8trip
   awplus(config-keychain-key)# accept-lifetime 08:30:00 Jan 09 2012 21:00:00 Feb 10 2012
   awplus(config-keychain-key)# send-lifetime 09:00:00 Jan 09 2012 20:00:00 Feb 10 2012
   awplus(config-keychain)# key 2
   awplus(config-keychain-key)# key-string hop78run
   awplus(config-keychain-key)# accept-lifetime 08:30:00 Feb 09 2012 21:00:00 Mar 10 2012
   awplus(config-keychain-key)# send-lifetime 20:30:00 Feb 09 2012 20:00:00 May 10 2012
   ```

3. Configure particular interfaces to use MD5 authentication of RIP, and specify the key chain to use in this authentication.

   ```
   awplus(config)# interface vlan1
   awplus(config-if)# ip rip authentication mode md5
   awplus(config-if)# ip rip authentication key-chain rip-key-chain
   ```

Best Practices for OSPF Configuration

OSPF is a link state routing protocol. It enables routers in a TCP/IP network to exchange information with each other to determine the best path for routing IP traffic to any given destination in the network. In essence, each router informs each other router about the state of its links – i.e. the interfaces via which it connects to its local subnets. The state of a link is a description of that interface and of its relationship to its neighboring routers.

A description includes its:

- IP address
- type of network it is connected to
- bandwidth-related 'cost'
- the identities of the neighboring routers

The link states are stored in a link state database, from which the best path to any destination can be derived.

The design of the OSPF protocol takes great account of its need to support network scalability. The operation of the protocol, and a number of features within the protocol, enable an OSPF network to grow very large without the amount of protocol traffic, and the amount of OSPF-related activity that routers must perform, growing uncontrollably. OSPF responds quickly to link failure and topology changes; this is particularly valuable in large networks where fast convergence is a priority.

Configure OSPFv2 with MD5 authentication

MD5 authentication is altogether more sophisticated than plan text authentication.

With MD5 authentication, OSPF does the following:

- Puts a packet sequence number into each packet. The sequence number is incremented each time a packet is transmitted on a given interface.
- Calculates the MD5 hash of the packet contents (including the sequence number), and a shared key.
- Puts this hash value into the packet.
- Puts an ID number for the shared key into the packet, so that recipients of the packet will know which, of potentially multiple, shared key has been used for the hash calculation in this packet.

By doing all this, OSPF achieves:

Authentication—the receiving router performs the same hash calculation, using its copy of the shared key. If the result that it gets from the hash calculation matches that which is in the packet, then that verifies that the sender is using the correct key.

Tamper-proofing—if the contents of the packet were altered along the way, then the result of the hash calculation performed by the recipient (using the tampered packet contents) will not match the hash calculated by the sender, even if they are both using the same key.

A guard against replay attacks—the monotonically incrementing sequence number means that an attacker cannot just store previously transmitted packets and replay them, as they would have an old sequence number. Also, because the sequence number is part of the packet contents used to calculate the hash, the attacker cannot just replace the sequence numbers in its stored packets by some higher sequence numbers, because the hash value would then fail to match the packet contents.

The value of including a key ID is that it enables neighbors to change shared keys, and choose to use different keys from the set at different times. This makes the cracking of the keys that much harder.

To configure OSPFv2 to use MD5 authentication, proceed as follows:

1. Initiate an OSPF routing process.

```
awplus(config)# router ospf 1
```

2. Configure the switch to use OSPF on the interfaces that fall within specified subnets.

```
awplus(config-router)# network 172.28.0.0 0.0.255.255 area 0
awplus(config-router)# network 221.189.62.0 0.255.255.255 area 4.3.3.1
```

3. Configure the areas to use MD5 authentication.

```
awplus(config-router)# area 0 authentication message-digest
awplus(config-router)# area 4.3.3.1 authentication message-digest
```

4. Leave OSPF configuration mode and then enter interface configuration mode to configure a message digest key per VLAN interface.

```
awplus(config-router)# exit
awplus(config)# interface vlan1
awplus(config-if)# ip ospf message-digest-key 1 md5 <key-string>
```

Configure OSPFv3 with authentication and encryption

Authentication and encryption in OSPFv3 uses the inherent authentication and encryption capabilities of the IPv6 protocol, and so operate quite differently to the authentication process in OSPFv2 .

In OSPFv3, you define security policies, that are applied to the OSPF configuration on individual interfaces.

A security policy consists of an:

- SPI (Security Parameter Index), which is simply an ID number for the policy
- authentication algorithm (if the policy is to perform authentication)
- authentication key (if the policy is to perform authentication)
- encryption algorithm (if the policy is to perform encryption)
- encryption key (if the policy is to perform encryption)

The available authentication algorithms are MD5 and SHA.

The available encryption algorithms are AES and 3DES.

The commands to configure OSPFv3 authentication on an interface are:

```
awplus(config)# interface vlan1
awplus(config-if)# ipv6 ospf authentication ipsec spi <256-4294967295> {md5 <MD5-key>|sha1 <SHA1-key>}
```

The commands to configure OSPFv3 encryption on an interface are:

```
awplus(config)# interface vlan1
awplus(config-if)# ipv6 ospf encryption ipsec spi <256-4294967295> esp {aes-cbc <AES-CBC-key>|3des <3DES-key>|null}
```

Both OSPFv3 encryption and authentication can be configured together on an interface with the commands:

```
awplus(config)# interface vlan1
awplus(config-if)# ipv6 ospf encryption ipsec spi <256-4294967295> esp {aes-cbc <AES-CBC key>|3des <3DES-key>|null}> {md5 <MD5-key>|sha1 <SHA1-key>}
```

Unless the OSPF network is small, divide it into Areas

One of the key benefits of OSPF is its scalability. This scalability is based on the fact that an OSPF network can be divided into Areas.

Areas limit the set of routers to which Router and Network LSAs are flooded. Imposing this limit keeps control of the total amount of OSPF packet exchange that goes on in the network. It also reduces the size of the LSA databases within individual routers - they don't need to hold the Router LSAs and Network LSAs originated by routers that are outside their Area.

For OSPFv2 (IPv4), under AlliedWare Plus, an Area is not explicitly created on a router. Rather, it is introduced implicitly when a network is configured, as the Area that the network belongs to has to be specified in that configuration command:

```
awplus# configure terminal
awplus(config)# router ospf 100
awplus(config-router)# network 10.0.0.0/8 area 1.1.1.1
```

Similarly, Areas are implicitly defined in OSPFv3 (IPv6) configuration. The difference in OSPFv3 is that the configuration is done on an interface, rather than a network.

To configure OSPFv3 on an interface, first enter interface configuration mode for that interface, then enter the command:

```
awplus(config-if)# ipv6 router ospf area <area-id> [tag <process-id>][instance <inst-id>]
```

The Area referred to by `<area-id>` does not need to have been pre-defined. Simply specifying an `<area-id>` in an **ipv6 router OSPF** command makes the router aware of that Area.

Configure route summarization at area boundaries

Route summarization has two main benefits:

1. The most obvious is that it is a way to reduce the size of route tables. This reduces the amount of memory that the routers require in order to store the routing tables, and the amount of computing power they need in order to perform route lookups at an acceptable rate.

2. Another, not quite so obvious, benefit is that summarization leads to less updates to route tables. Links at the edge of the network have a tendency to go up and down rather often. This results in the subnets at the edge changing between an available state and an unavailable state.

   When a given subnet changes between available and unavailable, routers are obliged to advertise this fact. However, if a router is not advertising every individual subnet that is downstream of it, but is advertising a summarized route, then the sudden unavailability of one particular subnet does not mean the whole summarized route is unavailable. So, there is no need for the summarizing router to make any change in what it advertises to upstream routers.

This reduces the amount of OSPF traffic that is exchanged. It also reduces the amount of time the routers spend recalculating routes and updating their route tables. In particular, if there is a route flapping (and interface going up and down with unseemly frequency) in a particular portion of the network, then summarization isolates the rest of the network from the negative effects of this flapping.

In order to take advantage of summarization, the subnets assigned to networks within a given Area should be assigned in a contiguous way, so they can be gathered together in a single range, or a small set of ranges.

The range(s) that encompass the subnets within an Area are configured on the Area. For example, if you want to summarize all the routes in Area 1 into the two summary routes 192.16.0.0/16 and 203.18.0.0/16, then the configuration is:

```
awplus# configure terminal
awplus(config)# router ospf 1
awplus(config-router)# area 1 range 192.16.0.0/16
awplus(config-router)# area 1 range 203.18.0.0/16
```

With this configuration, the router will advertise these two summary routes into the other areas that it borders. It will not advertise routes that are subnets within these summary routes.

Set the reference bandwidth to an appropriate value for your link speeds

By default, the OSPF interface cost is calculated as 10^8 /speed (in bits per second [bps]). This will result in calculating an OSPF cost of 1 for all interfaces faster than 100Mbps.

In a network that includes interfaces with bandwidths of 1Gbps, 10Gbps, or even 40Gbps, then these interfaces will all have the same cost. So, OSPF will consider a 100Mbps link just as desirable as a 40Gbps link, which is not a good way to direct traffic down the highest-bandwidth links.

The reference bandwidth feature allows flexibility in the OSPF cost assignments, without forcing you to manually assign OSPF costs to every interface. The reference bandwidth replaces the 10^8 component in the above formula. The reference bandwidth is specified in Mbps, i.e. the commands that follow specify the reference bandwidth to be 40Gbps, resulting in the calculation of OSPF interface costs:

```
awplus# configure terminal
awplus(config)# router ospf 100
awplus(config-router)# auto-cost reference-bandwidth 40000
```


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

alliedtelesis.com

C613-05044-00 REV A