Introduction

The OpenFlow protocol is a network protocol closely associated with Software-Defined Networking (SDN). SDN is a network architecture that allows network administrators to control traffic from a centralized Controller. A Controller is an application that manages flow control in an SDN environment. The OpenFlow protocol allows a server to instruct network switches where to send data packets.

In a non-OpenFlow or legacy switch, packet forwarding (the data path) and route determination (the control path) occur on the same device. A switch using the OpenFlow protocol separates the data path from the control path.

The OpenFlow protocol is used on the control plane (which is centralized on the SDN Controller) to communicate with the data plane (which is distributed among the network nodes) in an SDN network. Using the OpenFlow specifications, a switch can be configured to operate with similar results to a legacy switch, without having to manually re-configure the switch if the network changes.

The following AlliedWare Plus Series switches support the OpenFlow version 1.3 specification: x930, x510, DC2552XS/L3, x550, x310, x230. These switches enable the OpenFlow protocol on a per-port basis, so you can choose which ports of the switch will be controlled by the OpenFlow feature. Non-OpenFlow-enabled ports, continue to support existing features of the device.

An OpenFlow enabled port will handle all untagged and VLAN tagged traffic. A hybrid OpenFlow port allows some VLAN tagged traffic to be processed as non-OpenFlow protocol traffic. This is achieved by setting the port to trunk mode and adding VLANs to the port. Un tagged traffic and tagged traffic for all other VLANs are handled by the OpenFlow protocol.

The AT-Secure Enterprise SDN Controller (AT-SESC) is a component of the Allied Telesis SDN offering. AT-SESC is an SDN Controller, that uses the OpenFlow protocol to control AlliedWare Plus™ switches.
## Contents

Introduction................................................................................................................... 1

- Products and software version that apply to this guide............................................. 3

The OpenFlow Protocol Support Details ......................................................................... 4

- SDN Controllers and the OpenFlow protocol............................................................ 4
- Connecting devices to ports and table entry limits.................................................... 4
- Incompatibilities with other features......................................................................... 5
- Registering the OpenFlow protocol license key......................................................... 5

What is an OpenFlow Controller? ..................................................................................... 6

Communication and Packet Processing.............................................................................. 6

Security .................................................................................................................................. 7

- Commands.................................................................................................................. 7
- Configuration guidelines......................................................................................... 8

Configuring the Switch to use the OpenFlow Protocol..................................................... 10

- Common terms........................................................................................................ 10
- Commands........................................................................................................... 11
- Configuration guidelines..................................................................................... 11

Configuration Examples.................................................................................................... 12

- Example 1 - Configuring a switch to use the OpenFlow protocol......................... 12
- Example 2 - Configuring a switch with a hybrid port and AMF........................... 14

Understanding the Local Port............................................................................................ 17

Inactivity Timeout and Behavior....................................................................................... 19
**Products and software version that apply to this guide**

This guide applies to AlliedWare Plus™ products that support the OpenFlow protocol, running version 5.4.7 or later.

The following AlliedWare Plus Series switches support the OpenFlow version 1.3 specification: x930, x510, DC2552XS/L3, x550, x310, x230.

**AlliedWare Plus version 5.4.7 supports the following extensions to the OpenFlow protocol:**

1. A new type of OpenFlow port, the **hybrid** port, is supported. Hybrid ports allow for a number of VLANs on a port using OpenFlow technology, to be reserved for management purposes. Only tagged traffic on explicitly defined VLANs will be treated as legacy traffic, all other traffic will be treated as OpenFlow technology Controller traffic. Note that AMF traffic on specially reserved VLANs will be treated as legacy (that is, AMF) traffic, and not as OpenFlow protocol traffic.

2. The **local** port has been supported. This allows OpenFlow protocol rules with an input port or output port specified as Local. The purpose of this is to allow the OpenFlow protocol to control traffic to and from the network stack of the switches operating under the OpenFlow specification.

3. The local port manifests itself as an interface called "of0" in the switch. The of0 interface can have IP addresses assigned to it, and can also have sub-interfaces added to it based on VLAN ID.

**AlliedWare Plus version 5.4.7 also removes support for some features:**

1. The hairpin link is no longer supported. When upgrading from 5.4.6-2 or earlier to 5.4.7 or later, special care will have to be taken if a hairpin link is present. For advice on how to achieve this and minimize disruption, please contact Allied Telesis Support.

2. AMF guest nodes on ports using the OpenFlow protocol are no longer supported.

To see whether your product supports the OpenFlow protocol, 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
The OpenFlow Protocol Support Details

SDN Controllers and the OpenFlow protocol

The AT-SESC SDN Controller is available to control AlliedWare Plus switches in all markets with a variety of applications. AlliedWare Plus switches can also be used with third-party SDN Controllers that support version 1.0 and 1.3 of the OpenFlow protocol.

Connecting devices to ports and table entry limits

**x230 and x310 Series**

When using an x230 Series or x310 Series switch with AT-SESC, you should only connect one end-user device to each port using the OpenFlow protocol. When using an x230 Series or x310 Series switch with other Controllers, we recommend you apply the same limit of one end-user device per port.

**x930, x510, x510L, x550 Series and DC2552XS/L3**

When using an x930 Series, x510 Series, x510L Series, x550 Series, or DC2552XS/L3 switch, the maximum number of simultaneous active flows depends on the sizes of the products’ hardware flow tables. This is because active flows use ACLs.

The following table shows the maximum number of flow table entries available on each switch series:

<table>
<thead>
<tr>
<th>SWITCH SERIES</th>
<th>MAXIMUM NUMBER OF FLOW TABLE ENTRIES</th>
<th>APPROXIMATE MAXIMUM NUMBER OF END-USER DEVICES</th>
</tr>
</thead>
<tbody>
<tr>
<td>x930</td>
<td>2037</td>
<td>1000</td>
</tr>
<tr>
<td>x510</td>
<td>245</td>
<td>120</td>
</tr>
<tr>
<td>x510L</td>
<td>245</td>
<td>120</td>
</tr>
<tr>
<td>DC2552XS/L3</td>
<td>245</td>
<td>120</td>
</tr>
<tr>
<td>x550</td>
<td>511</td>
<td>255</td>
</tr>
<tr>
<td>x310</td>
<td>117</td>
<td>57</td>
</tr>
<tr>
<td>x230</td>
<td>117</td>
<td>57</td>
</tr>
</tbody>
</table>

When using AT-SESC, note that connections to end-user devices need two flow table entries. Therefore, the maximum number of devices you can connect is approximately half the number of flow table entries.

Also note that some SDN applications may require three or more flow table entries, per device, and that flow table entries may be used by other protocols. Both these factors may reduce the number of simultaneous flows that the switch can process.
x550 Series

The x550 Series support the OpenFlow protocol from software version 5.4.7-1.0 onwards. The maximum number of hardware flow table entries available on x550 product series is 511. Entries greater than 511 are processed in software. While using the OpenFlow protocol, we recommend not to use regular ACLs with the action ‘send-to-vlan’.

x510-52, and x310-50 Series

From software version 5.4.6-2.1 onwards, all ports on the x510-52 Series and x310-50 Series switches can be configured to use the OpenFlow protocol. On software versions prior to 5.4.6-2.1, you can choose ports from either port set 1 or port set 2, as shown in the following table:

<table>
<thead>
<tr>
<th>MODEL NAMES</th>
<th>PORT SET 1</th>
<th>PORT SET 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT-x510-52GTX</td>
<td>1.0.1-1.0.24</td>
<td>1.0.25-1.0.48</td>
</tr>
<tr>
<td>AT-x510-52GPX</td>
<td>1.0.50</td>
<td>1.0.49</td>
</tr>
<tr>
<td>AT-x510L-52GT</td>
<td>1.0.52</td>
<td>1.0.51</td>
</tr>
<tr>
<td>AT-x510L-52GP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AT-x510DP-52GTX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AT-x310-50FT</td>
<td>1.0.1-1.0.24</td>
<td>1.0.25-1.0.48</td>
</tr>
<tr>
<td>AT-x310-50FP</td>
<td>1.0.49</td>
<td>1.0.51</td>
</tr>
<tr>
<td></td>
<td>1.0.50</td>
<td>1.0.52</td>
</tr>
</tbody>
</table>

Incompatibilities with other features

Due to the way in which the OpenFlow protocol works, there is no guarantee that any legacy feature will work in conjunction with it. In particular, you cannot use the OpenFlow protocol together with the following features:

- VCStack
- Mirroring, on ports using the OpenFlow protocol
- Changing the egress queue or the internal priority of matching traffic on the ports connected to the OpenFlow Controller. Therefore, you cannot use the remark command on ports configured to use the OpenFlow protocol.

Registering the OpenFlow protocol license key

Before configuring AlliedWare Plus switches to use the OpenFlow protocol, you must obtain and register an OpenFlow protocol license key. Version 5.4.6-2 onwards adds support for OpenFlow protocol subscription licenses. To see the available licenses, check your device’s data sheet, which is available at alliedtelesis.com. Registering the OpenFlow protocol license key activates the OpenFlow feature on the switch.

To register the OpenFlow protocol license key, use the command:

```
awplus#license update file <bin-name>
```

As with most licensed features, it is recommended that the switch is rebooted before using the feature. See the Licensing Feature Overview and Configuration Guide for details.
What is an OpenFlow Controller?

An OpenFlow Controller is a software application that manages flow control in an SDN environment. Generally speaking, many SDN controllers are based on the OpenFlow protocol.

The OpenFlow Controller serves as a sort of operating system for the network. All communications between applications and devices have to go through the controller. The OpenFlow protocol connects the controller software to network devices so that server software can tell switches where to send packets for the forwarding table.

In this way, the controller uses the OpenFlow protocol to configure network devices to choose the best path for application traffic.

Communication and Packet Processing

There are two main things that occur in a switch using the OpenFlow protocol; they are communication with the Controller and packet processing:

1. Communication with the Controller
   - The switch has a Controller configured, and continuously attempts to connect to the Controller.
   - The Controller will ask the switch for status and statistics.
   - The Controller inserts OpenFlow specification flows on to the switch. These contain matches and actions (rules) that tell the switch what to do with packets. For example, a default rule might drop packets or send them to the Controller.

2. Packet processing
   Packets are processed either by:
   - Flows - as defined by the rules inserted by the Controller.

OR

- The CPU - software switched. Packets are passed through the rule tables and the net result (match and actions) for the packet's flow is discovered:
  - the flow is inserted into a software flow table (separate from the rule table)
  - packets are also software switched
  - if possible, the flow is added to a table in the switch silicon
  - subsequent packets in this flow will be switched by the hardware
if the flow cannot be added to the silicon, packets for the flow will be processed in software. The reasons for this include:

1. the flow table in silicon is full
2. actions cannot be executed by the silicon
3. chosen not to be processed

If the default rule is to drop, the flow can be added to silicon (to drop).

if the default rule is to send to the Controller, then the packet will be sent to the CPU.

Security

The switch to controller connection can be either TCP based, or SSL based. SSL is recommended for security, as the connection link is encrypted and authenticated. In order to set up a secure link, keys and certificates must be defined before the controller is added with the protocol specified as SSL.

Transport Layer Security (TLS) v1.0, TLS v1.1 and TLS v1.2 are supported on secure link(s). The TLS version used between an OpenFlow switch and OpenFlow Controller is determined by peer negotiation.

Commands

The commands to configure and monitor secure link(s) for the OpenFlow protocol are listed in the following table:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto pki trustpoint</td>
<td>Generates a unique private/public key pair and a certificate.</td>
</tr>
<tr>
<td>crypto pki export</td>
<td>Exports the CA certificate for its own certificate authority.</td>
</tr>
<tr>
<td>openflow ssl trustpoint</td>
<td>Specifies a trustpoint to be used for authentication.</td>
</tr>
<tr>
<td>openflow controller ssl</td>
<td>Connects to an OpenFlow Controller over TLS.</td>
</tr>
<tr>
<td>openflow ssl peer certificate</td>
<td>Changes validation mechanism of peer certificate on secure links(s) for the OpenFlow protocol.</td>
</tr>
<tr>
<td>show openflow ssl</td>
<td>Displays current SSL configuration for the OpenFlow protocol.</td>
</tr>
</tbody>
</table>
Configuration guidelines

To connect over TLS, every OpenFlow switch must have a unique private/public key pair and a certificate that signs the public key.

To create the key pair and certificate, follow the steps below:

**Step 1. Setup a local trustpoint**

```
awplus(config)#crypto trustpoint NAME
```

Where:
- **NAME** - the name of the local trustpoint to be set up. Note that only the 'local' trustpoint is supported as of 5.4.7-1. Once the 'local' trustpoint is set up, a 2048-bit RSA key and a self-signed certificate are created in either Flash or NVS, depending on whether secure mode is enabled or not on the OpenFlow switch. They will remain unless the user deletes the trustpoint with the `no` variant of the command.

**Step 2. Specify a trustpoint to authenticate the TLS encryption**

```
awplus(config)#openflow ssl trustpoint NAME
```

Where:
- **NAME** - the name of the trustpoint to be used for authentication.

**Step 3. Connect the OpenFlow switch to the OpenFlow Controller**

```
awplus(config)#openflow controller ssl A.B.C.D <1-65535>
```

Where:
- **A.B.C.D** - the IPv4 address of the OpenFlow Controller
- **<1-65535>** - the port number used to communicate with the OpenFlow Controller

**Step 4. Enable peer certificate validation** (disabled by default)

```
awplus(config)#openflow ssl peer certificate {FILEPATH|bootstrap}
```

Where:
- **FILEPATH** - the CA certificate for the controller(s)’ certificate authority.

Specify the path with an absolute path.
For example: flash:/certs/pki/local/cacert.pem. Download the certificate from the machine beforehand using a file `copy` command. Thereafter, the OpenFlow switch will only connect to OpenFlow Controller’s signed by the same CA certificate. The file must be PEM file format.

- **bootstrap** - specifies the bootstrap mode. The OpenFlow switch accepts and saves a self-signed certificate sent from the machine in which an OpenFlow controller is running. The OpenFlow switch obtains it from the machine on its first connection. Thereafter, the OpenFlow switch will only connect to OpenFlow Controllers signed by the same CA certificate.

**Note:** Peer certificate validation isn’t supported when secure mode is enabled with the `crypto secure-mode` command.
Step 5. Export the CA certificate for the OpenFlow Controller to validate

awplus#crypto pki export NAME pem {FILEPATH|terminal}

Where:

- **NAME** - the name of the trustpoint the CA certificate is to be exported
- **FILEPATH** - the URL that the PEM file is transferred to. The format of the URL is the same as any valid destination for a file `copy` command.
- **terminal** - the terminal to display the PEM file

Monitoring and managing configuration

To display the current SSL configuration, use the command:

awplus#show openflow ssl

```
awplus#show openflow ssl
Private key: /flash/.certs/pki/local/cakey.pem
Certificate: /flash/.certs/pki/local/cacert.pem
CA Certificate: /etc/openvswitch/cacert.pem
Bootstrap: true
```

To delete a trustpoint, use the command:

awplus(config)#no crypto pki trustpoint NAME

**Note:** It can only be deleted if TLS isn't used by an OpenFlow Controller connection(s).

To delete OpenFlow Controller settings, use the command:

awplus(config)#no openflow controller ssl A.B.C.D <1-65535>

To disable peer certificate validation, use the command:

awplus(config)#no openflow ssl peer certificate
Configuring the Switch to use the OpenFlow Protocol

This section includes a list of common terms, commands, and configuration guidelines when configuring a switch to use the OpenFlow protocol.

Common terms

Here is a brief description of some of the terms used in a scenario using the OpenFlow protocol:

- **Legacy port** - a port on the switch that is not controlled by the OpenFlow protocol, but instead by all the current (legacy) control protocols.

- **AMF Link** - an AMF link connects AMF capable devices, allowing them to join the AMF network.

- **Management port** - a management port cannot use the OpenFlow protocol and is best used just for managing the device.

- **OpenFlow port** - a port where data is controlled by rules obtained from a Controller using the OpenFlow protocol.

- **Hybrid port** - a port that behaves like an OpenFlow port for all traffic apart from traffic belonging to specifically configured VLANs, for which the traffic processing is like that of a legacy port.

- **Local port** - The local port enables remote entities to interact with the switch and its network services via the OpenFlow network, rather than via a separate control network. For more information about local ports, see "Understanding the Local Port" on page 17.
Commands

The commands for configuring and monitoring the OpenFlow feature are listed in the following table:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>openflow</td>
<td>Specifies a port to be under OpenFlow control</td>
</tr>
<tr>
<td>openflow controller</td>
<td>Specifies the OpenFlow Controller.</td>
</tr>
<tr>
<td>openflow version</td>
<td>Changes the supported OpenFlow protocol version number on the switch.</td>
</tr>
<tr>
<td>openflow native vlan</td>
<td>Specifies a native VLAN for the data plane ports.</td>
</tr>
<tr>
<td>show openflow config</td>
<td>Displays the OpenFlow protocol configuration from the configuration database.</td>
</tr>
<tr>
<td>show openflow coverage</td>
<td>Displays the counters from the OpenFlow protocol module in software.</td>
</tr>
<tr>
<td>show openflow flows</td>
<td>Displays the entries of the flow table on the switch.</td>
</tr>
<tr>
<td>show openflow rules</td>
<td>Displays the software flow table and rules set by the OpenFlow Controller.</td>
</tr>
<tr>
<td>show openflow status</td>
<td>Displays the status of each data plane port and OpenFlow protocol</td>
</tr>
</tbody>
</table>

For more information on these commands, see the product's Command Reference.

Configuration guidelines

To configure a switch to use the OpenFlow protocol:

- Obtain an OpenFlow protocol license.
- Apply the OpenFlow protocol license to the switch.
- Disable VCStacking.
- Create the VLAN used as the native VLAN for ports managed by the OpenFlow protocol. This VLAN must be different to the one used as the VLAN for the Control Plane.
- Set the IP address of the Control Plane.
- Configure the Controller for the OpenFlow protocol.
- Configure the native VLAN for the OpenFlow protocol.
  - Note, if the switch has both OpenFlow controlled ports and legacy ports, they need to have different native VLANs. You can change the native VLAN for either the OpenFlow controlled ports or the legacy ports.
- Enable the OpenFlow protocol.
- Disable RSTP and IGMP Snooping TCN Query Solicitation on the native VLAN for the OpenFlow ports.
- Set the IPv6 hardware filter size (if required)
Configuration Examples

Example 1 - Configuring a switch to use the OpenFlow protocol

This example uses an x510-28GTX switch. The following table lists the configuration details used in the examples below:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Control plane ports</strong></td>
<td>port1.0.1 to port1.0.4</td>
</tr>
<tr>
<td><strong>OpenFlow ports</strong></td>
<td>port1.0.5 to port1.0.28</td>
</tr>
<tr>
<td><strong>Native VLAN for Control Plane</strong></td>
<td>vlan1</td>
</tr>
<tr>
<td><strong>Native VLAN for OpenFlow ports</strong></td>
<td>vlan4089</td>
</tr>
<tr>
<td><strong>IP address for Control Plane interface</strong></td>
<td>192.168.1.1/24</td>
</tr>
<tr>
<td><strong>IP address of Controller</strong></td>
<td>192.168.1.10/24</td>
</tr>
<tr>
<td><strong>OpenFlow Controller Protocol</strong></td>
<td>TCP</td>
</tr>
<tr>
<td><strong>Controller port</strong></td>
<td>6633</td>
</tr>
</tbody>
</table>

**Figure 1: Pure OpenFlow protocol configuration**

**Step 1:** Apply the OpenFlow protocol license on the switch.

```
awplus#license update file <bin-name>
```

**Step 2:** Set the IP address of the Control Plane

```
awplus#configure terminal
awplus(config)#interface vlan1
awplus(config-if)#ip address 192.168.1.1/24
```
Step 3: Configure the Controller for the OpenFlow protocol.
awplus#configure terminal
awplus(config)#openflow controller tcp 192.168.1.10 6633

Step 4: Create the VLAN used as the native OpenFlow protocol VLAN. This VLAN must be different than the one used as the native for the Control Plane.
awplus#configure terminal
awplus(config)#vlan database
awplus(config)#vlan 4089

Step 5: Configure the Native VLAN for the OpenFlow protocol.
awplus(config)#openflow native vlan 4089

Step 6: Activate the ports controlled by the OpenFlow protocol
awplus#configure terminal
awplus(config)#interface port1.0.5-1.0.28
awplus(config-if)#openflow

Step 7: Disable RSTP and IGMP Snooping TCN Query Solicitation on the native VLAN for the OpenFlow protocol.
- The OpenFlow protocol requires that ports under its control do not send any control traffic, so you must disable RSTP and IGMP Snooping TCN Query Solicitation.
- Ensure there are no topology loops when RSTP is disabled.
awplus#configure terminal
awplus(config)#no spanning-tree rstp enable
awplus(config)#interface vlan4089
awplus(config-if)#no ip igmp snooping tcn query solicit

Step 8: Set the IPv6 hardware filter size (if required).
- Configure the following command if a packet is to be forwarded by IPv6 address matching.
- Please note that this command is supported on the x510 and x930 switches only. (The DC2552XS/L3 switch is not supported at this stage).
awplus#configure terminal
awplus(config)#platform hwfilter-size ipv4-full-ipv6
Example 2 - Configuring a switch with a hybrid port and AMF

This example describes how to configure an OpenFlow switch, with a hybrid port, and using AMF.

To recap, a hybrid port behaves like a port managed by the OpenFlow protocol for all traffic, apart from traffic belonging to specifically configured VLANs, for which the traffic processing is like that of a legacy port.

Figure 2: OpenFlow switch containing a hybrid port

The following table lists the configuration details used in the example and shown in Figure 3 below:

<table>
<thead>
<tr>
<th>X510-28GTX</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Control plane ports</td>
<td>port1.0.1 to port1.0.12</td>
</tr>
<tr>
<td>OpenFlow ports</td>
<td>port1.0.13 to port1.0.27</td>
</tr>
<tr>
<td>Hybrid OpenFlow port</td>
<td>port1.0.28</td>
</tr>
<tr>
<td>Tagged packets (VLANs) received on legacy port</td>
<td>vlan10</td>
</tr>
<tr>
<td>Native VLAN for Control Plane</td>
<td>vlan1</td>
</tr>
<tr>
<td>Native VLAN for the OpenFlow ports</td>
<td>vlan4089</td>
</tr>
<tr>
<td>IP address for the Control Plane interface</td>
<td>192.168.1.1/24</td>
</tr>
<tr>
<td>IP address of the Controller</td>
<td>192.168.1.10/24</td>
</tr>
<tr>
<td>OpenFlow Controller Protocol</td>
<td>TCP</td>
</tr>
<tr>
<td>AMF Network Name</td>
<td>Hybrid</td>
</tr>
<tr>
<td>AMF-link port</td>
<td>port1.0.28</td>
</tr>
</tbody>
</table>
Step 1: Configure the AMF network.

```
awplus#configure terminal
awplus(config)#atmf network-name Hybrid
```

Step 2: Apply the OpenFlow protocol license on the switch.

```
awplus#license update file <bin-name>
```

Step 3: Create a VLAN for the OpenFlow ports native VLAN.
- The OpenFlow ports native VLAN must be created before setting it.
- The VLAN ID for the native OpenFlow VLAN must be different from the native VLAN for the control plane

```
awplus#configure terminal
awplus(config)#vlan database
awplus(config-vlan)#vlan 4089
```

Step 4: Create a VLAN for native packets received on legacy ports

```
awplus(config-vlan)#vlan 10
```

Step 5: Configure the AMF link.

```
awplus#configure terminal
awplus(config)#interface port1.0.28
awplus(config-if)#switchport atmf-link
```

Step 6: Disable the ingress-filter for the hybrid port using the OpenFlow protocol to receive any untagged packets

```
awplus(config-if)#switchport mode trunk ingress-filter disable
```
Step 7: Assign a management VLAN to the hybrid port.
awplus(config-if)#switchport trunk allowed vlan add 1,10

Step 8: Enable this port to be managed by the OpenFlow protocol.
awplus(config-if)#openflow

Step 9: Set the IP address of the Control Plane.
awplus#configure terminal
awplus(config)#interface vlan1
awplus(config-if)#ip address 192.168.1.90/24

Step 10: Configure the OpenFlow protocol Controller.
awplus#configure terminal
awplus(config)#openflow controller tcp 192.168.1.2 6653

Step 11: Configure the native VLAN of the OpenFlow ports.
- You must set a dedicated native VLAN for OpenFlow ports.
- The OpenFlow native VLAN must be created before it is set.
- The VLAN ID for this native VLAN must be different from the VLAN for the Control Plane.

awplus#configure terminal
awplus(config)#openflow native vlan 4089

Step 12: Enable the ports to be managed by the OpenFlow protocol.
awplus#configure terminal
awplus(config)#interface port1.0.13-1.0.27
awplus(config-if)#openflow

Step 13: Disable RSTP and IGMP Snooping TCN Query Solicitation on the OpenFlow native VLAN.
- The OpenFlow protocol requires that ports under its control do not send any control traffic, so it is better to disable RSTP and IGMP Snooping TCN Query Solicitation.
- Ensure there are no topology loops when RSTP is disabled.

awplus#configure terminal
awplus(config)#no spanning-tree rstp enable

awplus(config)#interface vlan4089
awplus(config-if)#no ip igmp snooping tcn query solicit
Understanding the Local Port

The OpenFlow protocol has the concept of a reserved port number called **local**. The local port enables remote entities to interact with the switch and its network services via the OpenFlow protocol designed network, rather than via a separate control network.

The AlliedWare Plus implementation of the OpenFlow protocol supports the local port. The presence of the local port can be seen using the following `show` commands:

```
awplus#show openflow config
a904fb47-85af-48a3-8ed4-caec0c62938c
    Bridge "of0"
    ...
    Port "of0"
    Interface "of0"
    type: internal
```

**Note:** The bridge, port, and interface all have the same name "of0".

```
awplus#show openflow status
... LOCAL(of0): addr:02:a1:68:f5:59:65
    config: 0
    state: 0
    current: 10MB-FD
    speed: 10 Mbps now, 0 Mbps max
```

**Note:** The local port is not numbered, instead the keyword **LOCAL** is used. In all OpenFlow protocol interactions the number (0xffffffff) is used.

```
awplus#show interface of0
Interface of0
    Scope: both
    Link is UP, administrative state is UP
    Hardware is System tap
    IPv4 address 10.37.48.34/27 broadcast 10.37.48.63
    index 6 metric 1
    <UP,BROADCAST, RUNNING, MULTICAST>
    VRF Binding: Not bound
    SNMP link-status traps: Disabled
    Router Advertisement is disabled
    Router Advertisement default routes are accepted
    Router Advertisement prefix info is accepted
    input packets 72, bytes 7200, dropped 0, multicast packets 0
    output packets 0, bytes 0, multicast packets 0 broadcast packets 0
    Time since last state change: 0 days 03:01:55
```
Note that the MAC address for the interface is random and that it has local significance only (as opposed to being a globally assigned MAC address).

The basic of0 interface is for untagged traffic only. If you want to send tagged traffic to the local port, a VLAN tagged sub-interface has to be created.

Separate IP addresses can be added to the sub-interfaces.

In order for communication with the local port to work, the correct OpenFlow protocol rules must be put into the switch. The responsibility for this is with the OpenFlow Controller.
Inactivity Timeout and Behavior

The OpenFlow Controller manages the operation of switch port status and flows.

If the connection between the switch and controller is broken, or there are no controllers defined, you can configure the switch to behave in one of two ways: standalone or secure mode.

**Standalone mode**

To configure the switch for **standalone** mode, use the command:

```
awplus(config)#openflow failmode standalone
```

In standalone mode, if no message is received from the OpenFlow Controller for three times the inactivity probe interval, then the OpenFlow protocol will take over responsibility for setting up flows. The OpenFlow protocol will cause the switch to act like an ordinary MAC-learning switch, but continue to retry connecting to the controller in the background. When the connection succeeds, it will discontinue its standalone behavior.

**Secure mode**

To configure the switch for **secure** mode (which is also the default mode of operation), use the command:

```
awplus(config)#no openflow failmode standalone
```

In secure mode, the OpenFlow protocol will not set up flows on its own when the controller connection fails or when there are no controllers defined. The switch will continue to retry connecting to any defined controllers forever.

**Inactivity Timeout**

To control how long it will take for the switch to consider its connection to the controller broken, use the command:

```
awplus(config)#openflow inactivity <timeout>
```

Where `<timeout>` is the number of seconds before the switch will send an inactivity probe. The switch will wait two times the inactivity time before considering that the link has failed. The default inactivity probe timeout is 10s.