Technical Guide

The OpenFlow™ Protocol

Feature Overview and Configuration Guide

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.

A selection of Allied Telesis switches work with version 1.3 of the OpenFlow™ specification. 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. Untagged traffic and tagged traffic for all other VLANs are handled by the OpenFlow protocol.

The **AMF Sec Controller** (previously called AT-SESC) is a component of the Allied Telesis SDN offering. The AMF Sec Controller is an **SDN Controller**, that can use OpenFlow to control AlliedWare Plus™ switches.

Contents

Products and software version that apply to this guide

This guide applies to AlliedWare Plus™ products that support the OpenFlow protocol, running software version **5.4.7-0.x** or later.

To see whether your product supports OpenFlow, see the following documents:

- The product's Datasheet
- The product's Command Reference

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

Any configured trustpoint, not just the local self-signed trustpoint is supported from 5.5.2-1.x onwards:
A trustpoint is a named set of files including a private key and signed certificate that allows secure connection using SSL.

The hairpin link is supported until 5.4.6-2.x and from 5.4.9-0.x onwards:
For advice on whether or not to use the hairpin link, please contact Allied Telesis Support.

The following OpenFlow extensions are supported from 5.4.7-0.x onwards:

1. A new type of OpenFlow port, the **hybrid** port, is supported. Hybrid ports allow 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 controlled 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.

AMF guest nodes on ports using the OpenFlow protocol no longer supported from 5.4.7-0.x.

For more information, 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 AMF Sec 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, such as Faucet, that support version 1.0 and 1.3 of the OpenFlow protocol.

Connecting devices to ports and table entry limits

x230, x230L GS970M and x310 Series

When using an x230, x230L, GS970M, or x310 Series switch with the AMF Sec Controller, you should only connect one end-user device to each port using the OpenFlow protocol. When using these switches with other Controllers, we recommend you apply the same limit of one end-user device per port.

x930, x510 and DC2552XS/ L3 Series

When using an x930, x510/x510L Series, x550 Series or the 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 1: Maximum flow table entries and end-user devices for switch series

SWITCH SERIES	MAXIMUM NUMBER OF FLOW TABLE ENTRIES	APPROXIMATE MAXIMUM NUMBER OF END-USER DEVICES
SBx908 GEN2	8183	4000
x950	8183	4000
x930	2037	1000
x530L	2560	256
x530, GS980MX	1280	256
x510, x510L, IX5	245	120
DC2552XS/L3	757	370
x550	511	255 (see "x550 Series" on page 5)
x310	117	57
x230, x230L, GS970M	117	57
GS900MX/MPX	117	57
XS900MX	245	120
IE300, IE510	245	120
IE340, IE340L	119	57
IE210L	117	57

When using the AMF Sec Controller, 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.

x230L and IE210L Series

From software version **5.4.8-1.1** onwards, the x230L and IE210L Series support the OpenFlow protocol. The maximum number of hardware flow tables entries available on the x230L and IE210L Series is 117.

SBx908 GEN2

The SBx908 GEN2 supports a maximum number of flow table entries of 8183.

x550 Series

Traditionally, on our OpenFlow products, when the OpenFlow daemon wants to install a hardware ACL for certain actions like push_vlan and pop_vlan, we make use of EGR_L3_INTF to create a hardware entry for every unique SrcMAC, DstMAC and OuterVLAN. Usually, the number of these L3 interfaces is **more than** the TCAM size, with an exception as seen on AT-x550 product lines.

AT-x550 products are limited by hardware to contain 512 EGR_L3_INTF while having the capability for 1024 TCAM table (ACL table). In this case, since we cannot change the maximum number of ACL entries to be the same as EGR_L3_INTF value, we enforce a limitation for OpenFlow entries on x550.

- For AT-x550, we recommend only 511 flows in the TCAM table, which would be approximately a maximum of 255 end-user devices.
- This limitation also means that while the TCAM can accept entries greater than 512, any further flows will not be installed in hardware, but instead be processed in software.
- We also recommend that you should not use a regular ACL with the action **send-to-vlan-port** as this action also uses an EGR_L3_INTF entry, which may be currently exhausted by 512 OpenFlow flows.

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:

MODEL NAMES	PORT SET 1	PORT SET 2
AT-x510-52GTX AT-x510-52GPX AT-x510L-52GT AT-x510L-52GP AT-x510DP-52GTX	1.0.1-1.0.24 1.0.50 1.0.52	1.0.25-1.0.48 1.0.49 1.0.51
AT-x310-50FT AT-x310-50FP	1.0.1-1.0.24 1.0.49 1.0.50	1.0.25-1.0.48 1.0.51 1.0.52

x530/x530L Series

From software version **5.4.9-1.1** onwards, the x530, x530L Series support the OpenFlow protocol using the AMF Sec Controller. The AMF Sec Controller does not have the same flow matching abilities and actions as version 1.3 of the OpenFlow protocol.This means that not all flow matching abilities and actions are supported.

From software version **5.4.9-2.1** onwards, the x530, x530L Series support **version 1.3** of Openflow Specification. This means that the x530/x530L Series OpenFlow switches are able to match on any of the following fields:

- Ethernet type
- MAC addresses
- IP protocol value
- IPv4 addresses
- IPv6 addresses
- VLAN ID
- ECN value
- DSCP value
- TCP flag
- Layer 4 source and destination port numbers
- ICMP code

The x530/x530L Series OpenFlow switches are able to alter any of the following fields on matched flows:

- MAC source and destination addresses
- VLAN ID
- IPv4 addresses
- IPv6 source or destination addresses (you are only able to set one of them at a time, i.e. source or destination address).
- CoS value
- DSCP value
- Egress Queue
- Layer 4 source and destination port numbers

However, there are some cases where flows are only processed in software due to hardware limitations:

- Matching on MPLS header
- Matching on 4 fields in IPv6 packets if the IPv6 header includes extension headers, other than a single Hop-by-Hop Option header
- Setting the MAC SA to any address other than the address of the current device for packets that do not have a valid IP/IPv6 header
- Adding/removing/modifying the MPLS header
- Setting the ECN field
- Setting an IPv6 source and destination address at the same time
- Setting an IPv6 Layer 4 port number
- Modifying a destination MAC address for multicast traffic
- Modifying an IP address, Layer 4 port number, or source MAC address for multicast and broadcast Traffic

The maximum number of hardware flow table entries available on the 28-port switches is 1280. Switches with 52 ports have twice the number of hardware flow tables entries - 2560.

There are three VLAN actions available to modify the VLAN TCI field on matched flows:

- Pop VLAN tag
- Push VLAN tag
- Set VLAN ID

The x530/x530L Series support a maximum of 896 different VLAN actions and 255 MAC source addresses other than the address of the current device, and each VLAN action or MAC source address can be shared among multiple flows.

Utilizing hardware resources

This section describes how to utilize hardware resources in detail on the x530/x530L Openflow switches.

The x530/x530L Series use port-based ingress ACLs to match OpenFlow flows with an action.

The maximum possible number of flow entries is different on 28 and 52-port devices. 52-port devices have two switch instances, whereas 28-port devices only have one. By default, there are 512 ingress ACLs available for each switch instance. You can switch among 512, 1024, and 1280 ACLs using the command - **platform acls-to-vlanclassifiers**.

To fully ultilize the maximum number of flow entries available on 52-port devices, you need to distribute their flow evenly between two switch instances. The following table shows all possible mapping between physical ports and switch instance on 52-port models:

Table 2: Mapping physical ports and switch instances - 52-port models only

MODEL NAMES	SWITCH INSTANCE 1	SWITCH INSTANCE 0
AT-x530-52GTXm	port1.0.1-1.0.24 and port1.0.41-1.0.44	port1.0.25-1.0.40 and port1.0.45-1.0.52
AT-x530-52GPXm	port1.0.1-1.0.24 and port1.0.41-1.0.44	port1.0.25-1.0.40 and port1.0.45-1.0.52
AT-x530L-52GTX	port1.0.1-1.0.24	port1.0.25-1.0.52
AT-x530L-52GPX	port1.0.1-1.0.24	port1.0.25-1.0.52
AT-x530DP-52GHXm	port1.0.1-1.0.24 and port1.0.41-1.0.44	port1.0.25-1.0.40 and port1.0.45-1.0.52

Flow entry consumption

This section describes flow entry consumption on the x530/x530L Series supporting the AMF Sec Controller.

1. Ingress ACLs on the x530/x530L Series have two types of format - IPv6 and non-IPv6. An ingress ACL can only be set to match on either IPv6 or non-IPv6 traffic flows.
2. IPv6 full-match is not supported on x530 Series switches by default, you must first enable IPv6 full-match using the platform command - **platform hwfilter-size ipv4-full-ipv6**. When this command is enabled on an x530 Series switch, each IPv6 flow consumes twice the amount of hardware resource (i.e. twice the entry size) compared to an IPv4 flow.

   So for example with the default settings, the default Openflow rule that is installed is translated into two entries: a v6 entry and a non-v6 entry. These entries are both the same size with default settings. When you enable **platform hwfilter-size ipv4-full-ipv6**, the default Openflow rule will still be translated into two entries, but the v6 entry will be twice the size of the non-v6 entry. So it acts like it has taken up **3** hardware entries in total, rather than only 2 in the default mode.

Ingress ACL consumption are shown as following:

- 2 default rules per Openflow hybrid port
- 2 hybrid rules per trunked VLAN per Openflow hybrid port
- 1 rule per Openflow flow per Ingress Openflow port

For example:

OpenFlow native VLAN configuration:

```
openflow native vlan 4090
```

OpenFlow Hybrid port configuration:

```
interface port1.0.1
 openflow
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 10
 switchport trunk native vlan 4090
 flowcontrol both
```

OpenFlow port configuration:

```
interface port1.0.2
 openflow
 switchport
 switchport mode access
 switchport access vlan 4090
 flowcontrol both
```

- Total number of default rules = 2 * 2 (port1.0.1 + port1.0.2) = 4
- Total number of hybrid rules = 2 * 1 (VLAN 10) = 2
- If two different flows are seen on port1.0.2 = 2

  Total number of Ingress ACL consumption = 4 (default rules) + 2 (hybrid rules) + 2 (flows) = 8

Note that IPv6 full-match is not supported on x530 switches by default, you must first enable IPv6 full-match using the platform command - **platform hwfilter-size ipv4-full-ipv6**. When **platform hwfilter-size ipv4-full-ipv6** is enabled on an x530 switch, each IPv6 flow consumes twice the amount of hardware resource compared to an IPv4 flow.

Header modifications

On x530/x530L switches, HA (Header Alteration) is used to support IP Header Modifications for IPv4 and IPv6 flows. There are a maximum number of 1000 IPv4/v6 HA entries available for flows to share, and these HA entries share the same hardware resource with ARPs. Therefore, if all hardware resource is consumed by ARPs, it will no longer be possible to modify IPv4/v6 Headers using hardware switching.

HA/ARP consumption formula: 6 * TotalNumberOfHa + 1.5 * TotalNumberOfArp = 6000

- **IPv4 HA** - It is possible for flows to change IPv4 source and destination addresses, source and destination port numbers, and MAC destination address using one single IPv4 HA entry
- **IPv6 HA** - Changing IPv6 source and destination addresses is supported in hardware, and flows can only change one of them at a time. If a flow is configured to have its source and destination IPv6 addresses changed at the same time, or its port number changed, then the flow is processed in software (software-switching).

Note that **ipv6 forwarding** is disabled by default on all platforms, users are required to enable IPv6 forwarding to allow Openflow to modify IPv6 flows properly.

Additionally, IP/IPv6 header and MAC SA are modified in hardware via Policy-based Routing. By default, there are 127 Policy-based Routes available on the x530/x530L switch. Users are recommended to use the new platform command **platform sdn-route-ratio enhanced** to increase the limit to 1024. On products that do not have Policy-based Routing enabled, IP/IPv6 header and MAC SA modifications are not supported.

Note: There is always a 1-to-1 mapping between a flow and a policy-based route entry. When a single policy-based route entry is used to modify IP/IPv6 addresses, Layer 4 port numbers, or MAC SA, the matched flow may get modified with common Layer 2 or IP/IPv6 headers.

For example, if you have defined a flow to match on any traffic with an action of modifying the IP/IPv6 header, then all traffic will be modified with a common Layer 2 header as a result of using the identical policy-based route entry. This behavior may be undesirable in certain cases.

Example Openflow flow - match on **any** IPv4 TCP traffic with an action of setting TCP source port to 1234

- IPv4 TCP Stream A - source MAC=00:06:07:08:09:0a; TCP source port=80
- IPv4 TCP Stream B - source MAC=00:06:07:08:09:0b; TCP source port=80
    - Resulting Stream C - source MAC=00:06:07:08:09:0a; TCP source port=1234

To work-around this issue, you are recommended to configure additional match field such as MAC SA to separate a single flow into multiple flows, so each flow will be associated with an individual policy-based route entry and gets modified correctly.

- Example Openflow flow A - match on source MAC=00:06:07:08:09:0a IPv4 TCP traffic with an action of setting TCP source port to 1234
- Example Openflow flow B - match on source MAC=00:06:07:08:09:0b IPv4 TCP traffic with an action of setting TCP source port to 1234
    - IPv4 TCP Stream A - source MAC=00:06:07:08:09:0a; TCP source port=80
    - IPv4 TCP Stream B - source MAC=00:06:07:08:09:0b; TCP source port=80
        - Resulting Stream C - source MAC=00:06:07:08:09:0a; TCP source port=1234
        - Resulting Stream D - source MAC=00:06:07:08:09:0b; TCP source port=1234

OpenFlow and IPv6

Operation modes

AlliedWare Plus switches have two modes of operation for IPv6 and IPv4 traffic:

- **ipv4-limited-ipv6** - for all types of IPv4 traffic with limited support for IPv6 traffic.
- **ipv4-full-ipv6** - for all types of IPv4 and IPv6 traffic.

For optimum performance of OpenFlow matching on all IPv6 parameters, we recommended you use **ipv4-full-ipv6** mode.

To change to this mode, use the commands:

```
awplus# configure terminal
awplus(config)# platform hwfilter-size ipv4-full-ipv6
```

Note: If you change the mode, you must save the configuration and reboot the device.

Packet matching

OpenFlow has the flexibility to match on various aspects of a packet, including MAC address, Layer 4 (L4) port numbers, IPv4/IPv6 addresses, and IPv6 proto fields.

For IPv6 support on OpenFlow traffic, there are some restrictions/limitations involved with the current implementation.

- Firstly, as with the traditional AlliedWare Plus implementation, you need to be mindful of the correct IPv4/IPv6 operation mode, as mentioned above.
- Secondly, the ability to match on different aspects of IPv6 traffic differs between OpenFlow targets.

The following table shows the match criteria for each product series:

- **Software** indicates that any packets matching those conditions will be processed in software instead of consuming a hardware ACL entry.
- **Hardware** indicates that packets matching those conditions will be processed using a hardware ACL entry.

Table 3: Match criteria for IPv6 OpenFlow traffic

	MATCH CRITERIA 1	MATCH CRITERIA 2	SBX908 GEN2/X950	DC2552XS	X SERIES
Ethertype - 0x86dd Full Mode ipv4-full-ipv6	IPv6 Src/Dst Address	None	Hardware	Hardware	Hardware
		L4 Src/Dst Port	Software	Software	Hardware
		IPv6 proto	Software	Hardware	Hardware
		Src/Dst MAC Address	Hardware	Software	Hardware
	L4 Src/Dst Port	None	Hardware	Software	Hardware
		IPv6 Src/Dst Address	Software	Software	Hardware
		IPv6 proto	Hardware	Software	Hardware
		Src/Dst MAC Address	Hardware	Hardware	Hardware
	IPv6 proto	None	Hardware	Hardware	Hardware
		IPv6 Src/Dst Address	Software	Hardware	Hardware
		L4 Src/Dst Port	Hardware	Software	Hardware
		Src/Dst MAC Address	Hardware	Software	Hardware
	Src/Dst MAC Address	None	Hardware	Hardware	Hardware
		IPv6 Src/Dst Address	Hardware	Software	Hardware
		L4 Src/Dst Port	Hardware	Hardware	Hardware
		IPv6 proto	Hardware	Software	Hardware

Additional notes on matching

Here are some additional useful points to refer to:

- You can use the **show platform classifier statistics utilization brief** command to check the current consumption of hardware ACL entries.
- Even while in ipv4-full-ipv6 mode, all matching criteria for IPv4 traffic will continue to work as ipv4-limited-ipv6 mode with the exception of Layer 4 Source/Destination ports.
- In ipv4-full-ipv6 mode, matching on IPv6 address along with Layer 4 Source/Destination or IPv6 proto matching criteria will result in software processing.
- SBx908 GEN2/x950 switches currently have 256 hardware ACL entries available for matching on IPv6 traffic when in ipv4-full-ipv6 mode.
- DC2552XS switches currently have 127 hardware ACL entries available for matching on IPv6 traffic when in ipv4-full-ipv6 mode.
- All other switch implementations divide the total hardware group into roughly half size when in ipv4-full-ipv6 mode.

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 processing is performed by:

1. **Flows defined in software** - with two different paths:
   - The **slow path** in which packets for a new flow are passed through the OpenFlow rule tables to determine how they should be processed.
   - The **fast path** in which the flow determined by the slow path is used to optimise software switching.
2. **Flows in the switch silicon** - which are switched at wire speed. These are the same flows as created by the slow path, but installed into silicon where possible.

Software switching works as follows:

- Pass a packet through the rule tables and discover the net result (match and actions) for the packet's flow.
- Insert the flow into a software flow table (separate from the rule table).
- Software switch the packet.
- Attempt to add the flow 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. actions cannot be executed by the silicon
    2. the flow is chosen not to be processed
    3. the flow table in silicon is full

In the event that the silicon table is full, performance is improved by optimizing the use of silicon entries. Two main algorithms are in play:

- When a hardware entry is removed (due to not being used for a period of time, about 10s), the busiest software flow is installed in its place.
- At regular intervals, the least busy hardware flow will be swapped with the busiest software flow (as long as the software flow is actually busier that the hardware flow)
  - 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:

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

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 trustpoint**

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

Where:

- **NAME** - the name of the trustpoint to be set up.
- Note that only the 'local' trustpoint is supported prior to release 5.5.2-1. Starting with release 5.5.2-1, all trustpoints are supported.

Once the trustpoint is set up with the above command, 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.

If TLS isn't used by OpenFlow controller connection(s), you can use the following command to remove a trustpoint:

```
awplus(config)#no openflow ssl trustpoint
```

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

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

Where:

- ***<controller-name>*** - a user-specified or auto-generated (in the case of legacy syntax) controller name
- **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: A trustpoint 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 <controller-name>
```

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 25.

Commands

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

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

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.
- Disable VCStacking.
- Apply the OpenFlow protocol license to the switch.
- 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)
- Disable Loop Protection.

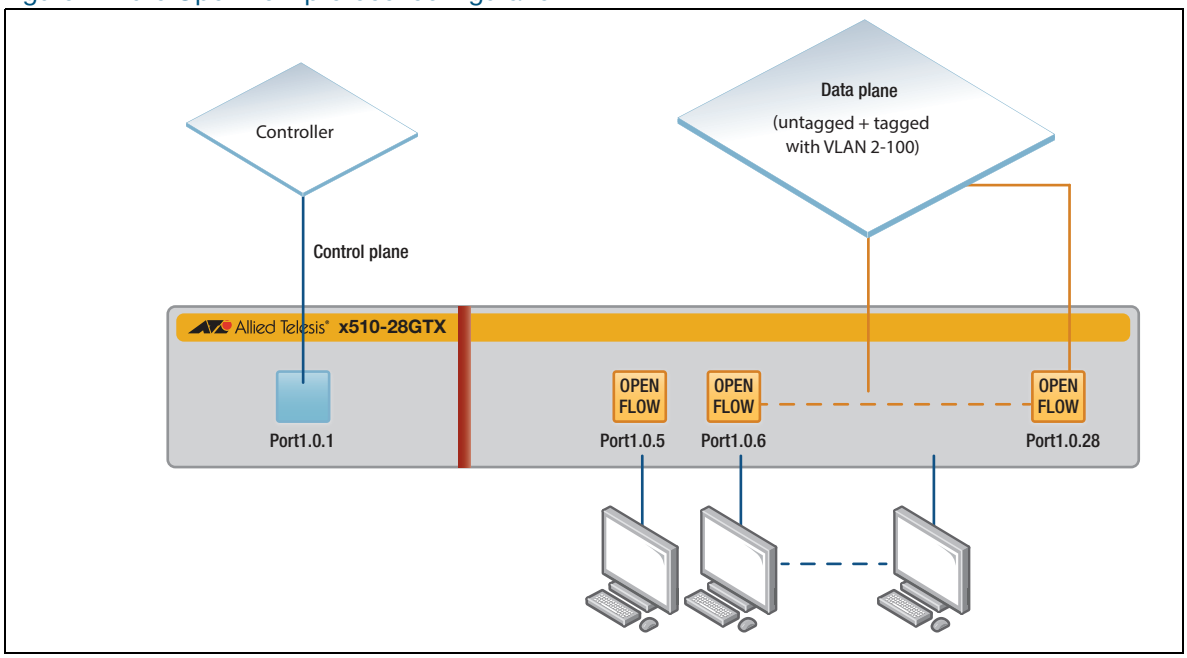
Configuration examples

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:

X510-28GTX	
Control plane ports	port1.0.1 to port1.0.4
OpenFlow ports	port1.0.5 to port1.0.28
Native VLAN for Control Plane	vlan1
Native VLAN for OpenFlow ports	vlan4089
IP address for Control Plane interface	192.168.1.1/24
IP address of Controller	192.168.1.10/24
OpenFlow Controller Protocol	TCP
Controller port	6653

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.**

- For our example, the name of the Controller is controller1. If you don't specify the controller name, the name is automatically created starting with 'ocxx' where xx is the sequential number starting from 1. e.g. oc1).

```
awplus#configure terminal
awplus(config)#openflow controller controller1 tcp 192.168.1.10 6653
```

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).**

```
awplus#configure terminal
awplus(config)#platform hwfilter-size ipv4-full-ipv6
```

Step 9. **Disable Loop Protection**

- The OpenFlow protocol requires that ports under its control do not send any control traffic. This means you should disable Loop Protection as well.

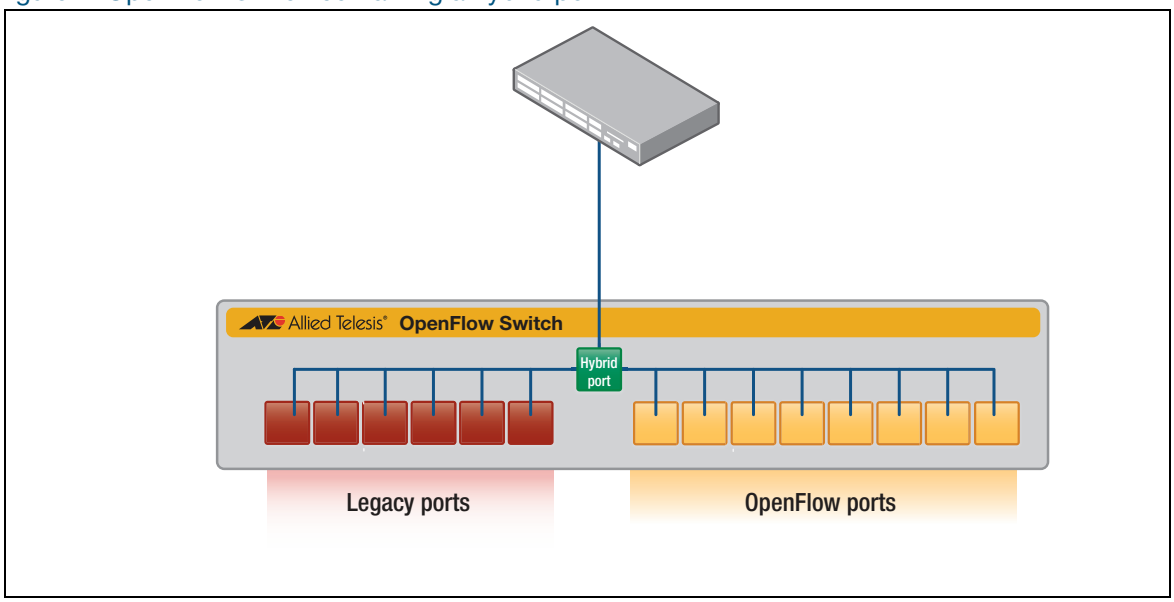
```
awplus#configure terminal
awplus(config)#no loop-protection loop-detect
```

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.

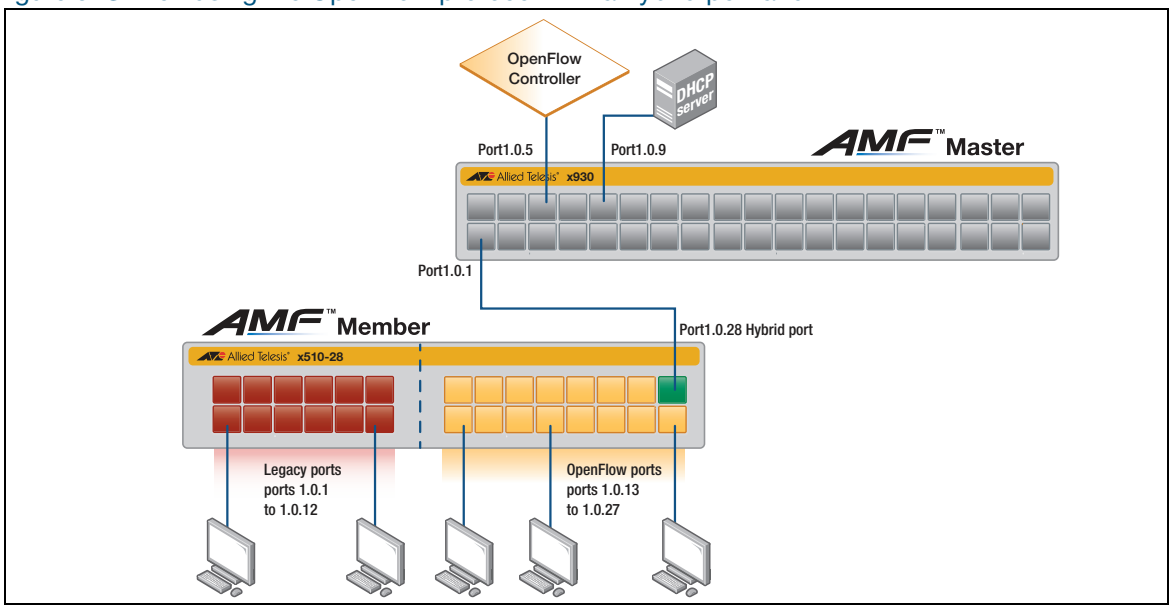
Figure 2: OpenFlow switch containing a hybrid port



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

X510-28GTX	
Control plane ports	port1.0.1 to port1.0.12
OpenFlow ports	port1.0.13 to port1.0.27
Hybrid OpenFlow port	port1.0.28
Tagged packets (VLANs) received on legacy (regular) port	vlan10
Native VLAN for Control Plane	vlan1
Native VLAN for the OpenFlow ports	vlan4089
IP address for the Control Plane interface	192.168.1.1/24
IP address of the Controller	192.168.1.10/24
OpenFlow Controller Protocol	TCP
AMF Network Name	Hybrid
AMF-link port	port1.0.28

Figure 3: Switch using the OpenFlow protocol with a hybrid port and AMF



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 (regular) 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: **Add the management VLAN(s) 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 192.168.1.1/24
```

Step 10: **Configure the OpenFlow protocol Controller.**

```
awplus#configure terminal
awplus(config)#openflow controller controller1 tcp 192.168.1.10 6653
```

- For our example, the name of the Controller is controller1. If you don't specify the controller name, the name is automatically created starting with 'ocxx' where xx is the sequential number starting from 1. e.g. oc1).

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 port1.0.13-1.0.28
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
```

Step 14. **Disable Loop Protection.**

- The OpenFlow protocol requires that ports under its control do not send any control traffic, so it's better to disable Loop Protection as well.

```
awplus#configure terminal
awplus(config)#no loop-protection loop-detect
```

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. With a suitable set of default flow entries it can be used to implement an in-band Controller connection, and defines an actual number for this port.

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 (0xfffffffe) 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
```

```
awplus#show run
!
interface of0
 ip address 10.37.48.34/27

awplus(config)#interface of0
awplus(config-if)#encapsulation dot1q 1234
awplus(config-if)#end
awplus#show interface of0.1234
Interface of0.1234
  Scope: both
  Link is UP, administrative state is UP
  Hardware is Encapsulated Ethernet, address is 6e41.b8ce.0382
  index 7 metric 1
  802.1Q VID 1234 over of0
  <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 0, bytes 0, dropped 0, multicast packets 0
    output packets 0, bytes 0, multicast packets 0 broadcast packets 0
  Time since last state change: 0 days 03:09:14
```

- 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.

```
awplus(config)#interface of0.1234
awplus(config-if)#ip address 10.37.48.121/27
```

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.

Note: If the OpenFlow switch is in fail mode, and the user changes the configured fail mode to or from standalone mode, OpenFlow will flush all existing rules

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 secure non-rule-expired
```

In secure mode, OpenFlow will not set up new flows on its own when the Controller connection fails or when no Controllers are defined, but all existing flows are left in place. The switch will continue to retry connecting to any defined Controllers forever. When the ***non-rule-expired*** option is enabled, existing rules won't be expired regardless of their timeouts while under fail-open mode. In other words, the OpenFlow switch will ignore timeout values of both idle timeout and hard timeout in existing rules.

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.

Hairpin link

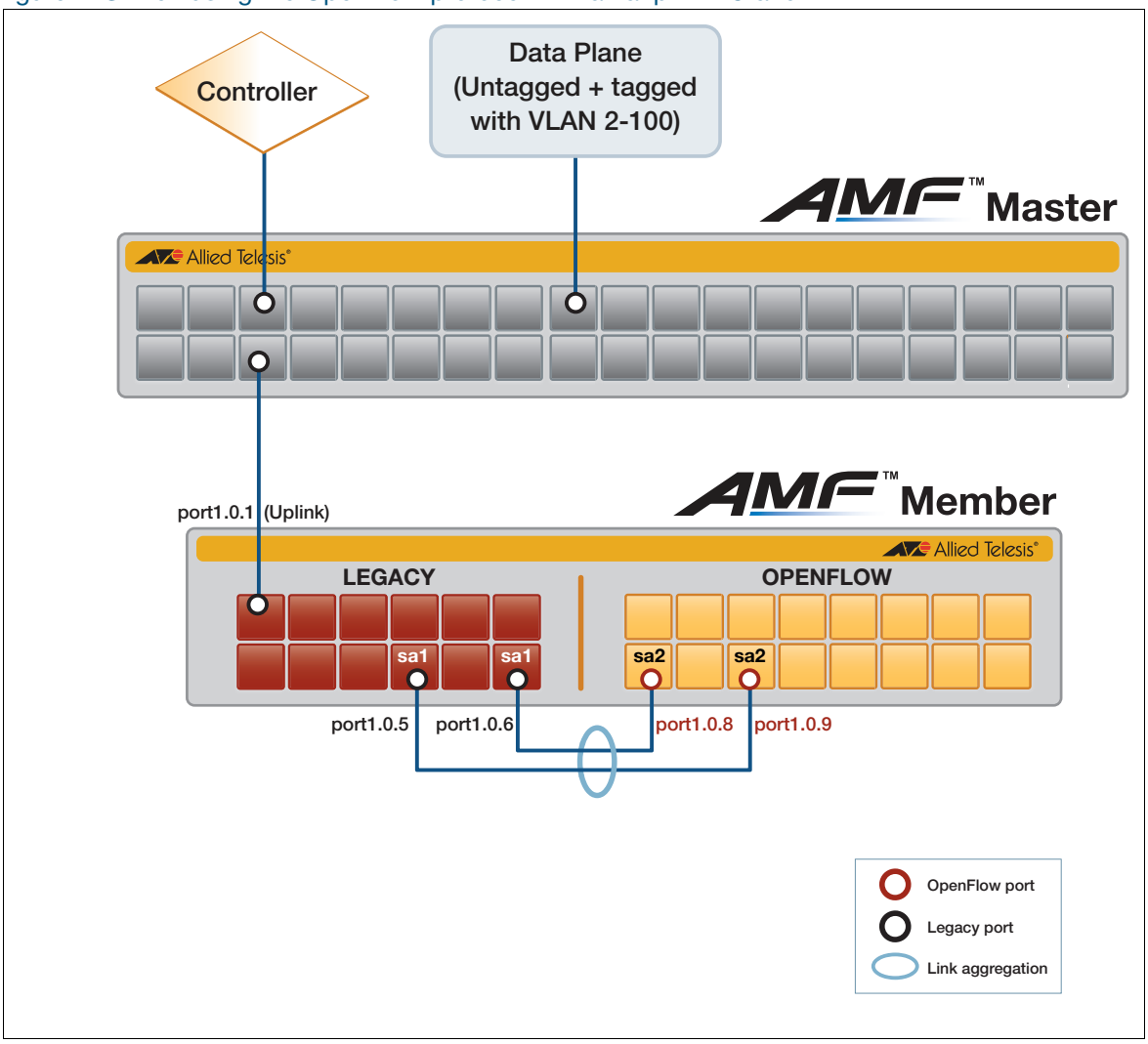
A hairpin link can be used to bridge the OpenFlow and legacy ports in a switch. The hairpin link is an alternative to hybrid ports, which are described earlier in this guide.

In a hairpin link, two switch ports (or a link aggregation group) are directly connected to each other. One of the ports is an OpenFlow port, the other a legacy port. This allows traffic to traverse between the OpenFlow controlled data plane and the legacy controlled data plane. The two ports used by the hairpin link can be referred to as hairpin ports.

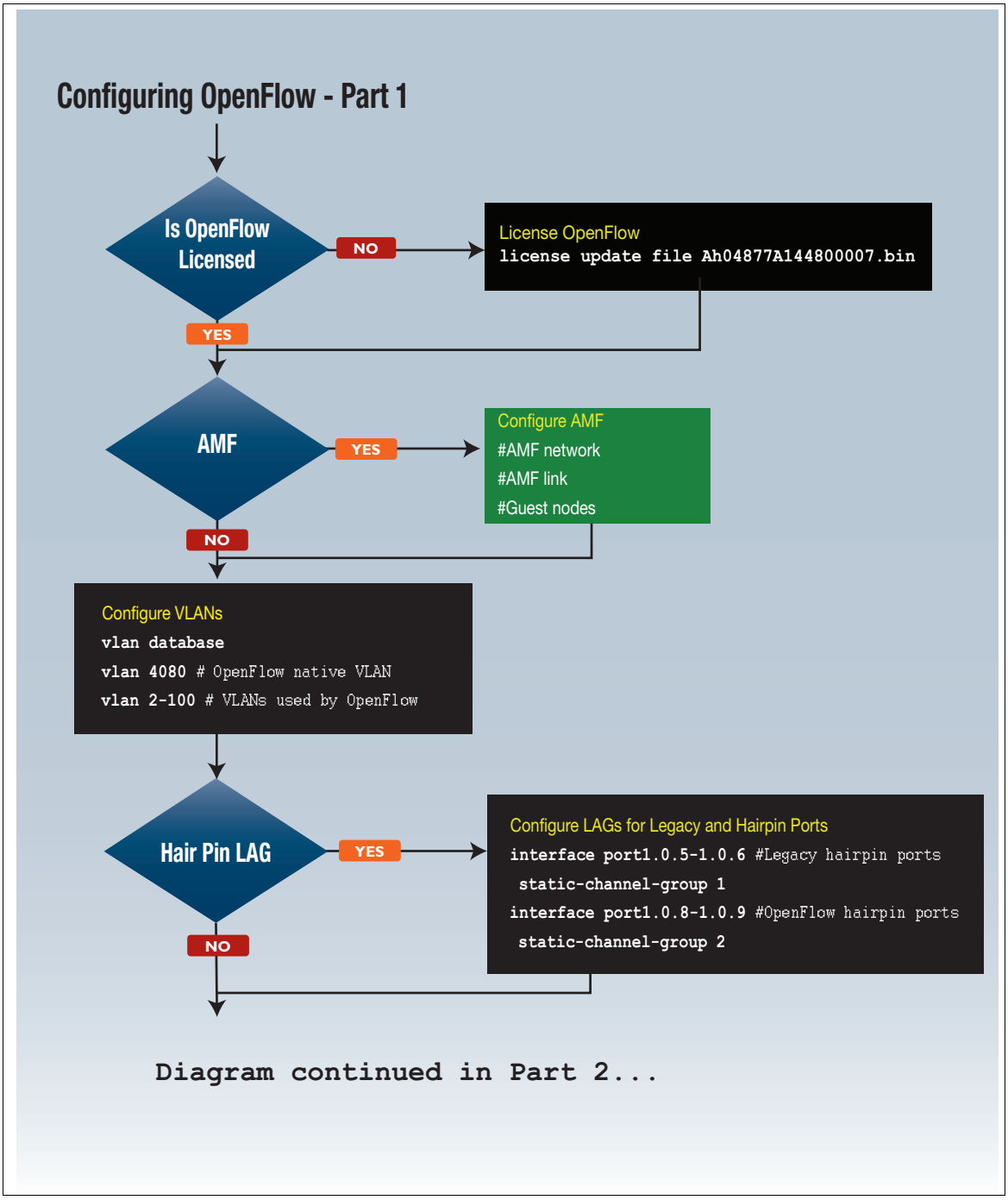
The hairpin link was supported up until v5.4.6-2.x and is supported again from v5.4.9-0.1 onwards, for situations where hybrid ports cannot be used. Because the hairpin link is a single point of failure, we recommend you discuss your network configuration with an Allied Telesis support representative before using it instead of hybrid ports. Note that 50- and 52-port switches do not support the hairpin link.

The following figure shows an example scenario with legacy and OpenFlow ports, connected by a hairpin link aggregation group (LAG):

Figure 4: Switch using the OpenFlow protocol with a hairpin LAG and AMF



The following two figures show how to configure the above example.



Configuring OpenFlow - Part 2

OpenFlow basic configuration

```
openflow native vlan <4080>
openflow controller <controller-name> tcp <192.168.1.1> <6653>
openflow version <1.0.1.3> #not required if version is just 1.3
```

Disable RSTP, IGMP Snooping TCN Query Solicit

```
no spanning-tree rstp enable
interface vlan2-100, vlan4080 #If interface required
 no ip igmp snooping tcn query solicit
```

Configure VLANs on Hairpin and Uplink

```
interface port1.0.5, port1.0.1 #hairpin, uplink
interface sa1, port1.0.1 #sa1 if Hairpin LAG used
 switchport mode trunk
 switchport trunk allowed vlan add 2-100
```

Configure OpenFlow Hairpin

```
interface port1.0.8 #if Hairpin LAG not used
interface sa2       #if Hairpin LAG used
  no ip igmp trusted query
  no ip igmp trusted routermode
```

Enable OpenFlow Ports

```
interface port1.0.8-1.0.28
 openflow
interface sa2 #if HPLAG is Yes
 openflow
```

Save config and possibly reboot switch

E.g. if you have just licensed OpenFlow, a reboot is required

C613-22084-00 REV S

Allied Telesis NETWORK SMARTER

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