

Openflow Custom Pipeline

Contents

Background	1
Standard pipeline	1
Switch Openflow configuration	2
Standard pipeline show Openflow.....	3
Show instance	3
Custom Pipeline	4
Custom Pipeline switch configuration	5
Custom pipeline Show instance	5
SDN Controller Topology	7
Network Visualizer Topology	7
Figure 1: Switch summary standard pipeline.....	2
Figure 2: Standard pipeline default rules.....	2
Figure 3: VAN default custom pipeline	4
Figure 4: Custom pipeline summary	5
Figure 5: Custom Pipeline Flows	5
Figure 6: SDN controller with mixed pipelines	7
Figure 7: Network Visualizer Topology	7
Figure 8: User data pushed from Clearpass.....	8
Figure 9: User mode capture	8
Figure 10: Custom Mode capture	9

Background

The purpose of this document is to demonstrate the support of custom OpenFlow pipelines in the ArubaOS switches and VAN controller. This capability has been supported in the switches since v16.01 and is now supported on the controller version 2.8. This is not a technical reference, it is simply an overview of custom pipelines and some screen grabs of the configuration and flow displays. This also shows the Aruba SDN apps running on custom pipelines as well as Clearpass – Visualizer integration.

Standard pipeline

Standard pipeline in VAN SDNC 2.8.8 and ArubaOS 16.03.0004

In the standard pipeline 3 tables are used 2 Hardware (0,100) and one Software table (200). With the controller default rules, table 0 simply points at table 100 so is not very useful. The switch only has a single hardware table (100) and it merely reports table 0 to be spec compliant. If we

attempt to push any rules to table 0, they are rejected. Table 100 copies BDDP topology discovery packets to the controller. It also matches on ARP and DHCP and forwards those packets to table 200. This works OK and is backward compatible with V2 ASIC based hardware. However there are only 2 tables that can be used for matching and applying actions. Furthermore table 200 is processed in the CPU so has limited performance and high overhead. The net result is that we only have one HW table for match and set actions. This is rather limiting and doesn't take full advantage of the V3 ASIC capabilities.

Summary for Data Path ID: 00:1f:b0:5a:da:98:31:60			
		Summary	Ports
Manufacturer:	Aruba	Data Path ID:	00:1f:b0:5a:da:98:31:60
H/W Version:	2930F-8G-PoE+-2SFP+ Switch	Address:	192.168.1.254
S/W Version:	WC.16.03.0004	Port:	56130
Serial #:	CN69HKZ01X	Negotiated Version:	1.3.0
Description:	sdntest	# Tables:	3
		# Buffers:	0
Capabilities			
flow_stats			
table_stats			
port_stats			
group_stats			
port_blocked			

Figure 1: Switch summary standard pipeline

Flows for Data Path ID: 00:1f:b0:5a:da:98:31:60

Summary

Ports

Flows

Groups

Table ID	Flow Count	Table Name				
▼ 0	1	Start				
▶ Priority 0	Packets 0	Bytes 0	Match	Actions/Instructions goto_table: 100	Flow Class ID com.hp.sdn.normal	
▼ 100	5	Policy Table				
▶ Priority 60000	Packets 0	Bytes 0	Match eth_type: bddp	Actions/Instructions apply_actions: output: CONTROLLER	Flow Class ID com.hp.sdn.bddp.steal	
▶ 31500	0	0	eth_type: ipv4 ip_proto: udp udp_src: 67 udp_dst: 68	goto_table: 200	com.hp.sdn.dhcp.copy	
▶ 31500	0	0	eth_type: ipv4 ip_proto: udp udp_src: 68 udp_dst: 67	goto_table: 200	com.hp.sdn.dhcp.copy	
▶ 31000	0	0	eth_type: arp	goto_table: 200	com.hp.sdn.arp.copy	
▶ 0	71	0		apply_actions: output: NORMAL	com.hp.sdn.normal	
▼ 200	4	SW Table 1				
▶ Priority 31500	Packets 0	Bytes 0	Match eth_type: ipv4 ip_proto: udp udp_src: 67 udp_dst: 68	Actions/Instructions apply_actions: output: CONTROLLER output: NORMAL	Flow Class ID com.hp.sdn.dhcp.copy	
▶ 31500	0	0	eth_type: ipv4 ip_proto: udp udp_src: 68 udp_dst: 67	apply_actions: output: CONTROLLER output: NORMAL	com.hp.sdn.dhcp.copy	
▶ 31000	0	0	eth_type: arp	apply_actions: output: CONTROLLER output: NORMAL	com.hp.sdn.arp.copy	
▶ 0	0	0		apply_actions: output: NORMAL	com.hp.sdn.normal	

Figure 2: Standard pipeline default rules

Switch Openflow configuration

```
Aruba-2930F-8G-PoEP-2SFP(openflow)# display this
controller-id 1 ip 192.168.1.3 controller-interface vlan 1
instance "sdntest"
member vlan 31-33
controller-id 1
version 1.3 only
connection-interruption-mode fail-standalone
default-miss-action output-normal
```

enable
exit
enable

Standard pipeline show Openflow

Aruba-2930F-8G-PoEP-2SFPP# show openflow

OpenFlow : Enabled
Egress Only Ports Mode : Disabled

Instance Information

Instance Name	No. of Oper.	Status	No. of H/W Flows	OpenFlow S/W Flows	Version
sdntest	Up	6	4	1.3 only	

Show instance

Aruba-2930F-8G-PoEP-2SFPP# show openflow instance sdntest

Configured OF Version : 1.3 only
Negotiated OF Version : 1.3
Instance Name : sdntest
Data-path Description : sdntest
Administrator Status : Enabled
Member List : VLAN 31-33

Pipeline Model : **Standard Match**
Listen Port : None
Operational Status : Up
Operational Status Reason : NA
Datapath ID : 001fb05ada983160
Mode : Active
Flow Location : Hardware and Software
No. of Hardware Flows : 6
No. of Software Flows : 4
Hardware Rate Limit : 0 kbps
Software Rate Limit : 100 pps
Conn. Interrupt Mode : Fail-Standalone
Maximum Backoff Interval : 60 seconds
Probe Interval : 10 seconds
Hardware Table Miss Count : NA
No. of Software Flow Tables : 1
Egress Only Ports : None
Table Model : Policy Engine and Software
Source MAC Group Table : Disabled
Destination MAC Group Table : Disabled
Default Miss Action : Output-Normal
Packet-In VLAN Tagging : Default

Controller Id Connection Status Connection State Secure Role

1 Connected Active No Equal

Custom Pipeline

Since the 16.01 firmware release, the v3 switch models (5400, 3810, 2930) have supported a customisable OpenFlow pipeline. The customisable pipeline provides the following improvements over prior OpenFlow implementations:

1. Optimization of available hardware resources, for improved scaling.
2. Multi-app coordination, using multiple successive hardware tables.
3. Improved functionality, with matches/instructions only available with a customized pipeline

SDNC 2.8.8 and the Aruba SDN applications (Protector, Optimizer and Visualizer) can now use the custom pipeline. This creates multiple tables and shift the rules to HW. On the switch this is signalled with the command 'Custom Pipeline' under the Openflow instance. This is dynamically negotiated with the controller so can be initialised on a switch by switch basis. This provides backward compatibility and flexibility to apply custom pipeline(s) on a switch by switch basis. Aruba has a default custom pipeline definition that works with our SDN Apps. You can create your own pipeline and push this down to the switch. There is a tool available to build these pipelines for VAN and RYU (eg FAUCET) controllers. See tools appendix

When a switch is configured with "custom pipeline-model" and connected to the VAN SDN controller, the controller will push a default customized pipeline, referred to as the "default VAN custom pipeline" elsewhere in this document. The default VAN custom pipeline is general purpose, yet gives improved scaling for HPE VAN applications (Protector, Optimizer, Visualizer).

The default VAN custom pipeline is show in the following table

Table ID	Name	Max entries	Purpose
0	Base Table	2	Enable/disable pipeline processing
10	Visibility	520	Copy user traffic
20	Infrastructure Bypass	2048	Prevent traffic from infrastructure ports from matching tables 30,40,50. Examples include ARP and DNS traffic.
30	Topology	8	Discover topology by copying user traffic
40	Security	420	Inspect user traffic, block malicious hosts
50	QoS/Optimization	520	Modify QoS on user traffic
100	General	512	General-purpose functionality not available in other tables
250	Default Forwarding	2	Default forwarding behavior, based on hybrid.mode setting

Figure 3: VAN default custom pipeline

Summary for Data Path ID: 00:1f:b0:5a:da:98:31:60			
		Summary	Ports
Manufacturer:	Aruba		
H/W Version:	2930F-8G-PoEP-2SFPP+ Switch		
S/W Version:	WC.16.03.0004		
Serial #:	CN69HKZ01X		
Description:	sdntest		
Data Path ID:	00:1f:b0:5a:da:98:31:60		
Address:	192.168.1.254		
Port:	61564		
Negotiated Version:	1.3.0		
# Tables:	1		
# Buffers:	0		
Capabilities			
flow_stats			
table_stats			
port_stats			
group_stats			
port_blocked			

Figure 4: Custom pipeline summary

Flows for Data Path ID: 00:1f:b0:5a:da:98:31:60			
		Summary	Ports
Table ID	Flow Count	Table Name	
▼ 0	1	Base	
► Priority	Packets	Bytes	Match
► 0	0	0	0
Actions/Instructions	goto_table: 10		
Flow Class ID	com.hp.sdn.normal		
▼ 10	2	Visibility	
► Priority	Packets	Bytes	Match
► 60000	0	0	eth_type: bddp
Actions/Instructions	apply_actions: output: CONTROLLER		
Flow Class ID	com.hp.sdn.bddp.steal		
► 0	0	0	0
goto_table: 20			
com.hp.sdn.normal			
▼ 20	2	Infrastructure Bypass	
► Priority	Packets	Bytes	Match
► 34000	0	0	in_port: 8
Actions/Instructions	goto_table: 30		
Flow Class ID	com.hp.sdn arp.filter		
► 0	0	0	0
goto_table: 30			
com.hp.sdn.normal			
▼ 30	4	Topology	
► Priority	Packets	Bytes	Match
► 31500	0	0	eth_type: ipv4
Actions/Instructions	write_actions: output: CONTROLLER		
Flow Class ID	com.hp.sdn.dhcp.copy		
► 31500	0	0	ip_proto: udp
write_actions:	output: CONTROLLER		
com.hp.sdn.dhcp.copy			
► 31500	0	0	udp_src: 67
goto_table: 40			
com.hp.sdn.dhcp.copy			
► 31500	0	0	udp_dst: 68
write_actions:	output: CONTROLLER		
com.hp.sdn.dhcp.copy			
► 31000	0	0	eth_type: arp
write_actions:	output: CONTROLLER		
com.hp.sdn.arp.copy			
► 0	0	0	0
goto_table: 40			
com.hp.sdn.normal			
▼ 40	1	Security	
► Priority	Packets	Bytes	Match
► 0	0	0	0
Actions/Instructions	goto_table: 50		
Flow Class ID	com.hp.sdn.normal		
▼ 50	1	QoS/Optimization	
► Priority	Packets	Bytes	Match
► 0	0	0	0
Actions/Instructions	goto_table: 100		
Flow Class ID	com.hp.sdn.normal		
▼ 100	1	General	
► Priority	Packets	Bytes	Match
► 0	0	0	0
Actions/Instructions	goto_table: 250		
Flow Class ID	com.hp.sdn.normal		
▼ 250	1	Default Forwarding	
► Priority	Packets	Bytes	Match
► 0	0	0	0
Actions/Instructions	apply_actions: output: NORMAL		
Flow Class ID	com.hp.sdn.normal		

Figure 5: Custom Pipeline Flows

Custom Pipeline switch configuration

```
Aruba-2930F-8G-PoEP-2SFPP(of-inst-sdntest)# disp this
instance "sdntest"
member vlan 31-33
controller-id 1
version 1.3 only
connection-interruption-mode fail-standalone
pipeline-model custom
default-miss-action output-normal
enable
exit
```

Custom pipeline Show instance

```
Aruba-2930F-8G-PoEP-2SFPP(of-inst-sdntest)# sh openflow
```

```
OpenFlow           : Enabled
Egress Only Ports Mode : Disabled
```

Instance Information

Instance Name	Oper. Status	No. of H/W Flows	No. of S/W Flows	OpenFlow Version
sdntest	Up	13	0	1.3 only

Aruba-2930F-8G-PoEP-2SFPP(config)# show openflow instance sdntest

Configured OF Version : 1.3 only
 Negotiated OF Version : 1.3
 Instance Name : sdntest
 Data-path Description : sdntest
 Administrator Status : Enabled
 Member List : VLAN 31-33

Pipeline Model : Custom Pipeline
 Listen Port : None
 Operational Status : Up
 Operational Status Reason : NA
 Datapath ID : 001fb05ada983160
 Mode : Active
 Flow Location : Hardware Only
 No. of Hardware Flows : 13
 No. of Software Flows : 0
 Hardware Rate Limit : 0 kbps
 Software Rate Limit : 100 pps
 Conn. Interrupt Mode : Fail-Standalone
 Maximum Backoff Interval : 60 seconds
 Probe Interval : 10 seconds
 Hardware Table Miss Count : NA
 No. of Software Flow Tables : NA
 Egress Only Ports : None
 Table Model : Custom Pipeline
 Source MAC Group Table : Disabled
 Destination MAC Group Table : Disabled
 Default Miss Action : Output-Normal
 Packet-In VLAN Tagging : Default

Controller Id Connection Status Connection State Secure Role

1	Connected	Active	No	Equal
---	-----------	--------	----	-------

SDN Controller Topology

The SDN controller is showing a topology with a 2930 V3 switch running Custom Pipeline and a 5400zl switch running standard pipeline. You can even run custom pipeline and standard pipeline on the same switch with different instances.

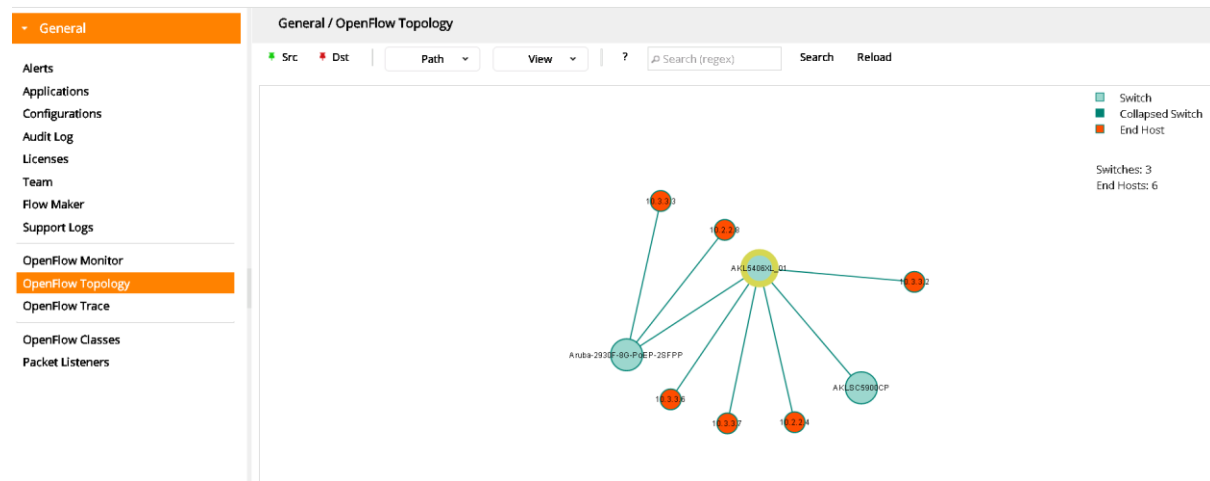


Figure 6: SDN controller with mixed pipelines

Network Visualizer Topology

The Visualizer application integrates with Active Directory (with AD agent) or with Aruba Clearpass. This provides User ID to the application. In this case integration is via Clearpass and we can see the end users that have associated with 802.1x. We can then create a capture session based on userID.

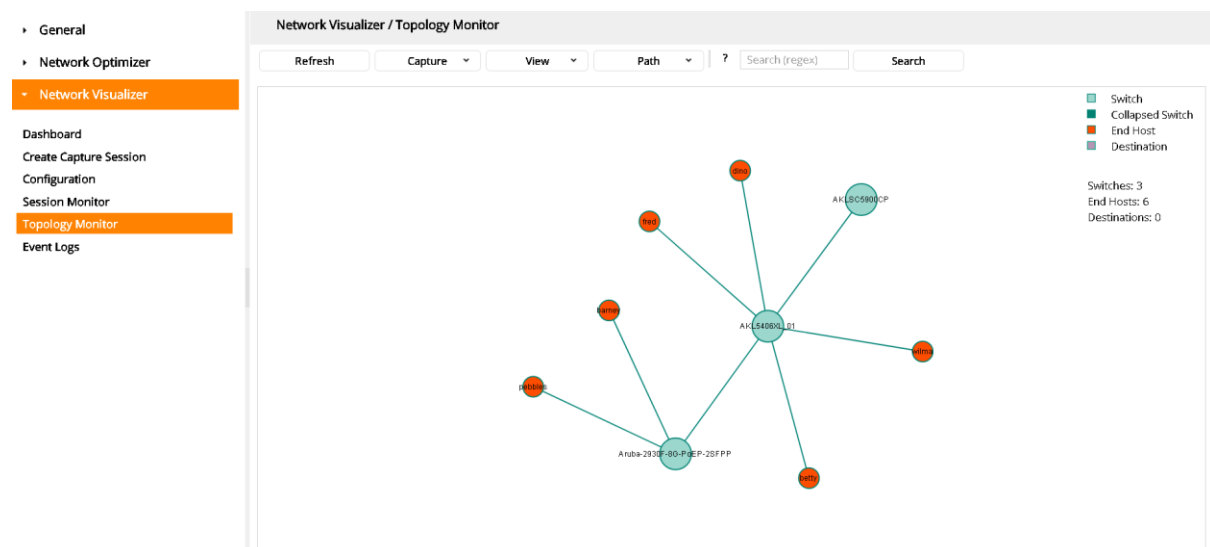


Figure 7: Network Visualizer Topology

Network Visualizer / Configuration

Applications
Users

Configured users.

	Name	Group
<input type="radio"/>	bambam	unknown
<input checked="" type="radio"/>	barney	unknown
<input type="radio"/>	betty	unknown
<input type="radio"/>	dino	unknown
<input type="radio"/>	fred	unknown
<input type="radio"/>	george	unknown
<input type="radio"/>	pebbles	unknown
<input type="radio"/>	wilma	unknown

Delete
Import
Refresh
Devices

Name

MAC Address (aa:bb:cc:dd:ee:ff)

Add

Clear

Configured user devices.

<input type="checkbox"/>	IP Address	MAC Address	Vendor	OS	Type
<input type="checkbox"/>	10.3.3.3	00:0f:61:01:46:f2	-	-	laptop/desktop

Delete

Figure 8: User data pushed from Clearpass

Clearpass is configured to push the end user information (UID, MAC) to Visualizer on authentication. The Clearpass integration is covered in the Visualizer Administration guide

Network Visualizer / Create Capture Session

Reset

Session Name
Filter Policy
Destination
Schedule
Summary
Status

This wizard walks you through the steps for configuring the capture session. You can navigate to different steps by clicking on the left panel.

Session Name

Session Mode
☒ User
☐ Custom

User Mode : Select User, User Group, Devices or Application

Figure 9: User mode capture

Network Visualizer / Create Capture Session

Reset

Session Name

Filter Policy

Destination

Schedule

Summary

Status

This wizard walks you through the steps for configuring the capture session. You can navigate to different steps by clicking on the left panel.

Session Name

Session Mode ☐ User ☒ Custom

Custom Mode: Select Protocol, Source and Destination Ports, IP/MAC Addresses

Figure 10: Custom Mode capture

Additional benefits

The custom pipeline implementation greatly increases the OpenFlow scalability on a v3 switch (2K flows to 8K Flows). It also permits multi-app coordination that was not possible with the standard pipeline. For example you should be able to have Protector inspect DNS and Optimizer set QoS for the same user.

All in all custom pipeline unlocks lots of hidden potential in the Aruba switch hardware and greatly extends the supported use cases.

References

[Aruba VAN SDN Controller 2.8 Custom Pipeline Support Guide](#)

[Aruba VAN SDN Controller 2.8 Installation Guide](#)

[Aruba VAN SDN Controller 2.8 Administrator Guide](#)

[HPE Network Visualizer SDN Application 1.1 Administrator Guide](#)

Tools

Thanks to Shaun Wackerly we have a couple of really useful tools for development and testing.

The first is a flowbuilder tool that will generate JSON



flow.html

The second is custom pipeline builder



pipeline.html