Application Visibility and Control

- Application visibility and control, on page 1
- Default DSCP on AVC, on page 11
- AVC-based selective reanchoring, on page 14
- AVC-based selective reanchoring, on page 21
- Restrictions for AVC-based selective reanchoring, on page 21
- Configure the flow exporter (CLI), on page 21
- Configure the flow monitor (CLI), on page 22
- Configure the AVC reanchoring profile (CLI), on page 23
- Configure the wireless WLAN profile policy (CLI), on page 23
- Verify AVC reanchoring, on page 25

Application visibility and control

Application visibility and control is a wireless network feature set that

- enables real-time identification and monitoring of applications using deep packet inspection
- allows creation of policy rules to manage application bandwidth and usage, and
- integrates with Flexible NetFlow (FNF) to report traffic statistics per application or protocol.

Application visibility and control (AVC) is a subset of the Flexible NetFlow (FNF) package. AVC provides traffic information. The AVC feature uses a distributed approach that benefits from NBAR running on the AP or controller. The goal is to run deep packet inspection (DPI) and report the results using FNF messages.

The NBAR2 engine analyzes and recognizes traffic flows. Each specific flow is marked with the detected protocol or application. You can use this per-flow information for application visibility through FNF. After application visibility is established, you can define control rules with policing mechanisms for a client.

Guidelines for application visibility and control

- Using AVC rules, bandwidth for a particular application can be limited for all clients joined on the WLAN. These bandwidth contracts coexist with per-client downstream rate limiting. The per-client limits take precedence over the per-application rate limits.
- The FNF feature is supported in wireless deployments. It relies on NetFlow enablement on the controller for all modes: FlexConnect, local and Fabric.

- In local mode, NBAR runs on the controller hardware. Client traffic flows through the data plane of the controller using the AP CAPWAP tunnels.
- In FlexConnect or Fabric mode, NBAR runs on the AP. Only statistics are sent to the controller . In these two modes, APs regularly send FNFv9 reports to the controller . The FNF feature uses these reports to provide application statistics reported by AVC.
- In Fabric mode, FNF cache is not populated. The system relays FNFv9 reports as they arrive. As a result, some flow monitor configuration parameters, such as cache timeout, are not used.

The behavior of the AVC solution changes based on the wireless deployment mode. The next sections describe the commonalities and differences in all scenarios.

Local Mode

- NBAR is enabled on the controller .
- AVC does not push the FNF configuration to the APs.
- Roaming events are ignored.

  However, AVC supports Layer 3 roaming in local mode because traffic flows through the anchor controller, where NBAR first processed the client's traffic when the client joined.
- IOSd needs to trigger NBAR attach.
- The solution supports flow monitor cache and NetFlow exporter.

FlexConnect Mode

- NBAR is enabled on an AP.
- AVC pushes the FNF configuration to the APs.
- Context transfer for roaming is supported in AVC-FNF.
- Flow monitor cache is supported.
- NetFlow exporter is supported.

Fabric Mode

- NBAR is enabled on an AP.
- AVC pushes the FNF configuration to the APs.
- Context transfer for roaming is supported in AVC-FNF.
- Flow monitor cache is not supported.
- NetFlow exporter is supported. For the C9800 on Catalyst switches for SDA, FNF cache is not available on the device.

Prerequisites for application visibility and control

- The APs should be AVC capable.

However, this requirement does not apply in Local mode.

- Configure the application visibility feature with FNF to enable AVC control (QoS).

Restrictions for application visibility and control

- IPv6 packet classification, including ICMPv6 traffic, is not supported in FlexConnect mode or Fabric mode. It is supported in Local mode.
- Layer 2 roaming is not supported across controller .
- Multicast traffic is not supported.
- AVC is supported only on the following APs:
  - Cisco Catalyst 9100 Series Access Points
  - Cisco Aironet 1800 Series Access Points
  - Cisco Aironet 2700 Series Access Point
  - Cisco Aironet 2800 Series Access Point
  - Cisco Aironet 3700 Series Access Points
  - Cisco Aironet 3800 Series Access Points
  - Cisco Aironet 4800 Series Access Points
  - Cisco Industrial Wireless 3702 Access Point
- AVC is not supported on Cisco Aironet 702W, 702I (128 M memory), and 1530 Series APs.
- Only applications recognized by App Visibility can be used for applying QoS control.
- Data link is not supported for NetFlow fields in AVC.
- You cannot map the same WLAN profile to both the AVC-not-enabled policy profile and the AVC-enabled policy profile.
- AVC is not supported on the management port (Gig 0/0).
- NBAR-based QoS policy configuration is allowed only on wired physical ports. Policy configuration is not supported on virtual interfaces. For example, VLAN, port channel and other logical interfaces.
- NBAR cannot classify traffic accurately when SaaS applications use end-to-end encryption, QUIC, or DoH because encryption affects classification. In these cases, encrypted traffic including DoH and QUIC without SNI prevents NBAR from sending the correct Protocol ID, which causes traffic classification issues.

When AVC is enabled, the profile supports a maximum of 23 rules, including the default DSCP rule. If the number of rules exceeds 23, the AVC policy will not be sent to the AP.

Configure AVC

Enable monitoring and policy enforcement for application usage on the wireless network.

Procedure

Step 1 Create a flow monitor using the **record wireless avc basic** command.
Step 2 Create a wireless policy profile.
Step 3 Apply the flow monitor to the wireless policy profile.
Step 4 Create a wireless policy tag.
Step 5 Map the WLAN to the policy profile.
Step 6 Attach the policy tag to the APs.

Create a flow monitor (CLI)

The NetFlow configuration requires a flow record, a flow monitor, and a flow exporter. Make this configuration your first step in the overall AVC configuration.

Note In FlexConnect mode and Local mode, the default values for **cache timeout active** and **cache timeout inactive** commands do not provide optimal AVC performance. Set both values to 60 in the flow monitor.

For Fabric mode, the cache timeout configuration does not apply.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a flow monitor.

Example:

```
Device(config)# flow monitor monitor-name fm_avc
```

Step 3 Specify the basic IPv4 or IPv6 wireless AVC flow template.

Example:

```
Device(config-flow-monitor)# record wireless avc {ipv4 | ipv6} basic
```

Note
If you want to have both Application Performance Monitoring (APM) and AVC-FNF in the device simultaneously, use the **record wireless avc {ipv4 | ipv6} assurance** command, which is a superset of the fields contained in **record wireless avc {ipv4 | ipv6} basic** command. If the containing flow monitor is configured with the local exporter using **destination wlc local**command, AVC-FNF will populate the statistics exactly as that of the **record wireless avc {ipv4 | ipv6} basic** configuration. As a result, both APM and AVC-FNF can be configured simultaneously with two flow monitors per direction, per IP version, in local (central switching) mode.

Note
The **record wireless avc basic** command is same as **record wireless avc ipv4 basic** command. However, **record wireless avc ipv4 basic** command is not supported in FlexConnect or Fabric modes. In such scenarios, use the **record wireless avc basic** command.

Step 4 Set the active flow timeout in seconds.

Example:

```
Device(config-flow-monitor)# cache timeout active value 60
```

Step 5 Set the inactive flow timeout in seconds.

Example:

```
Device(config-flow-monitor)# cache timeout inactive value 60
```

Configure a flow monitor (GUI)

Set up a flow monitor and export data to a collector using a pre-configured flow exporter.

Before you begin

Ensure you have already created a flow exporter.

Procedure

Step 1 Choose **Configuration** > **Services** > **Application Visibility** and go to the **Flow Monitor** tab.

Step 2 In the **Monitor** area, click **Add** to add a flow monitor.

Step 3 In the **Flow Monitor** window, add a flow monitor and a description.

Step 4 Select the Flow exporter from the drop-down list to export data from the flow monitor to the collector.

Note
To export wireless NetFlow data, use these templates:

- ETA (Encrypted Traffic Analysis)
- wireless avc basic
- wireless avc basic IPv6

Step 5 Click **Apply to Device** to save the configuration.

Create a flow record (CLI)

The default flow record cannot be edited or deleted. If you require a new flow record, create one and map it to the flow monitor from the CLI.

Procedure

Step 1 Create a flow record.

Example:

```
Device(config)# flow record flow_record_name record1
```

Note
When a custom flow record is configured in FlexConnect and Fabric modes, the optional fields (fields that are not present in record wireless avc basic) are ignored.

Step 2 (Optional) Describe the flow record as a maximum 63-character string.

Example:

```
Device(config-flow-record)# description IPv4flow
```

Step 3 Specify a match to the IPv4 protocol and specify a match to the IPv4 source address-based field.

Example:

```
Device(config-flow-record)# match ipv4 protocol
Device(config-flow-record)# match ipv4 source address
```

Step 4 Specify a match to the IPv4 destination address-based field and specify a match to the transport layer's source port field.

Example:

```
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match transport source-port
```

Step 5 Specify a match to the transport layer's destination port field and specify a match to the direction the flow was monitored in.

Example:

```
Device(config-flow-record)# match transport destination-port
Device(config-flow-record)# match flow direction
```

Step 6 Specify a match to the application name.

Example:

```
Device(config-flow-record)# match application name
```

Note
This action is mandatory for AVC support because this allows the flow to be matched against the application.

Step 7 Specify a match to the SSID name identifying the wireless network and collect the counter field's total bytes.

Example:

```
Device(config-flow-record)# match wireless ssid
Device(config-flow-record)# collect counter bytes long
```

Step 8 Collect the counter field's total packets and collect the BSSID with the MAC addresses of the APs that the wireless client is associated with.

Example:

```
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# collect wireless ap mac address
```

Step 9 Collect the MAC address of the client on the wireless network.

Example:

```
Device(config-flow-record)# collect wireless client mac address
```

Create a flow exporter (CLI)

Define export parameters for a flow exporter so AVC statistics are visible on the controller embedded wireless controller. This is an optional procedure for configuring flow exporter parameters.

Note For the AVC statistics to be visible at the controller, you should configure a local flow exporter using these commands:

- **flow exporter** *my_local*
- **destination local wlc**

Also, your flow monitor must use this local exporter for the statistics to be visible at the controller .

Procedure

Step 1 Create a flow monitor.

Example:

```
Device(config)# flow exporter flow-export-name export-test
```

Step 2 Describe the flow record as a maximum 63-character string.

Example:

```
Device(config-flow-exporter)# description IPv4flow
```

Step 3

Example:

```
Device(config-flow-exporter)# destination {hostname/ipv4address | hostname/ipv6address |
local wlc}
```

Step 4 Specify the local controller to which the exporter sends data.

Example:

```
Device(config-flow-exporter)# destination {hostname/ipv4address | hostname/ipv6address |
local wlc}
```

Step 5 (Optional) Configure the destination UDP port to reach the external collector.

Example:

```
Device(config-flow-exporter)# transport udp port-value 1024
```

The default value is 9995.

Note
This step is required only for external collectors; not required for local controller collector.

Step 6 (Optional) Specify the application table timeout option, in seconds.

Example:

```
Device(config-flow-exporter)# option application-table timeout 500
```

The valid range is from one to 86400.

Step 7 Return to privileged EXEC mode.

Example:

```
Device(config-flow-exporter)# end
```

Step 8 (Optional) Verify your configuration.

Example:

```
Device# show flow exporter
```

Configure a policy tag (CLI)

Establish a policy tag to control wireless device and SSID behavior.

Policy tags allow you to associate specific policy profiles with APs, enabling customized wireless network behavior.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure policy tag and enter policy tag configuration mode.

Example:

```
Device(config-policy-tag)# wireless tag policy policy-tag-name rr-xyz-policy-tag
```

Step 3 Save the configuration and exit configuration mode and return to privileged EXEC mode. **end**

Example:

```
Device(config-policy-tag)# end
```

Attach a policy profile to a WLAN interface (GUI)

Link the appropriate policy configuration to a WLAN interface for centralized control. Use the GUI to ensure new or modified policy profiles are correctly associated with WLANs before deployment.

Procedure

Step 1 Choose **Configuration** > **Tags & Profiles** > **Tags**.

Step 2 On the **Manage Tags** page, click the **Policy** tab.

Step 3 Click **Add** to open the **Add Policy Tag** window.

Step 4 Enter a name and description for the policy tag.

Step 5 Click **Add** to map WLAN and policy.

Step 6 Select the WLAN profile to map with the appropriate policy profile, and click the tick icon.

Step 7 Click **Save & Apply to Device**.

Attach a policy profile to a WLAN interface (CLI)

Assign a policy profile to a WLAN interface to control traffic and services for that WLAN.

Procedure

Step 1 Create a policy tag.

Example:

```
Device(config)# wireless tag policy avc-tag
```

Step 2 Attach a policy profile to a WLAN profile.

Example:

```
Device(config-policy-tag)# wlan wlan_avc policy avc_pol
```

What to do next

- Do not attach different AVC policy profiles to the same WLAN across different policy tags.

  This is an example of incorrect configuration:

```
wireless profile policy avc_pol1
ipv4 flow monitor fm-avc1 input
ipv4 flow monitor fm-avc1 outputno shutdown
wireless profile policy avc_pol2
ipv4 flow monitor fm-avc2 input
ipv4 flow monitor fm-avc2 output
no shutdown
wireless tag policy avc-tag1
wlan wlan1 policy avc_pol1
```

```
wireless tag policy avc-tag2
wlan wlan1 policy avc_pol2
```

This example violates the restriction stated earlier, that is, the WLAN *wlan1* is mapped to 2 policy profiles, *avc_pol1* and *avc_pol2*. This configuration is incorrect because the WLAN *wlan1* must be mapped to either *avc_pol1* or *avc_pol2* throughout the configuration..

- Conflicting policy profiles on the same WLAN are not supported. For example, policy profile (with and without AVC) applied to the same WLAN in different policy tags.

  This is an example of an incorrect configuration:

```
wireless profile policy avc_pol1
no shutdown
wireless profile policy avc_pol2
ipv4 flow monitor fm-avc2 input
ipv4 flow monitor fm-avc2 output
no shutdown
wireless tag policy avc-tag1
wlan wlan1 policy avc_pol1
wireless tag policy avc-tag2
wlan wlan1 policy avc_pol2
```

  In this example, a policy profile with and without AVC is applied to the same WLAN in different tags.

- Run the **no shutdown** command on the WLAN after completing the configuration.
- If the WLAN is already in **no shutdown** mode, run the **shutdown** command, then run the **no shutdown** command.

Attach a policy profile to an AP (CLI)

Assign a specific policy profile to an AP through commands.

Procedure

Step 1 Enter AP configuration mode.

Example:

```
Device(config)# ap ap-ether-mac 34a8.2ec7.4cf0
```

Step 2 Specify the policy tag that is to be attached to the AP.

Example:

```
Device(config)# policy-tag avc-tag
```

Verify the AVC configuration (CLI)

Review and validate the Application Visibility and Control (AVC) configuration on a device.

Procedure

Step 1 Display information about the top applications and the users who use them.

Example:

```
Device# show avc wlan wlan_avc top 2 applications {aggregate | downstream | upstream}
```

Note
To collect accurate statistics, ensure that wireless clients are connected to the WLAN and transmitting data. After 90 seconds, run the command.

Step 2 Display information about the top applications for the client.

Example:

```
Device# show avc client 9.3.4 top 3 applications {aggregate | downstream | upstream}
```

Note
Ensure that wireless clients are connected to the WLAN and transmitting data. Wait 90 seconds for statistics to become available, then run the command.

Step 3 Display information about the top applications and the users who uses them.

Example:

```
Device# show avc wlan wlan_avc application app top 4 aggregate
```

Step 4 Display a summary of all the APs attached to the controller .

Example:

```
Device# show ap summary
```

Step 5 Display a summary of all the APs with policy tags.

Example:

```
Device# show ap tag summary
```

Default DSCP on AVC

Configuring default DSCP for AVC profile (GUI)

Assign a default DSCP value for an Application Visibility and Control (AVC) profile using the GUI.

Procedure

Step 1 Choose **Configuration** > **Services** > **QoS**.

Step 2 Click **Add**.

Step 3 Enter the **Policy Name**.

Step 4 Click **Add Class-Maps**.

Step 5 Select **AVC** from the **AVC/User Defined** drop-down list.

Step 6 Click either **Any** or **All** match type radio button.

Step 7 Select **DSCP** from the **Mark Type** drop-down list.

Step 8
a) Check the **Drop** check box to drop traffic from specific sources.
b) If you do not want to drop the traffic, enter the **Police(kbps)** and select the match type from the **Match Type** drop-down list. Choose the items from the available list and click move them to the selected list. Click **Save**.

Step 9 Click **Apply to Device**.

Guidelines to configure default DSCP marking for unclassified AVC packets

When using Cisco Catalyst 9800 Series Wireless Controller or , ensure you configure a default DSCP marking for packets not matched by AVC policy filters. This allows consistent QoS treatment, even when your policy exceeds the 32-filter limit or cannot individually classify certain packet types.

- Set a default marking action in your class map and policy map to handle all unclassified traffic.
- Apply this configuration to ensure unclassified packets receive the intended QoS treatment.

Create class map (CLI)

Define a class map to identify and categorize network traffic based on application names, categories, subcategories, or application groups.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a class map.

Example:

```
Device(config-pmap)# class-map avc-class
```

Step 3 Specify match to the application name, category name, subcategory name, or application group.

Example:

```
Device(config)# class-map avc-class
Device(config-cmap)# match protocol avc-media
Device(config)# class-map class-avc-category
Device(config-cmap)# match protocol attribute category avc-media

Device# class-map class-avc-sub-category
Device(config-cmap)# match protocol attribute sub-category avc-media
```

```
Device# class-map avcS-webex-application-group
Device(config-cmap)# match protocol attribute application-group webex-media
```

Step 4 Return to privileged EXEC mode. Alternatively, press Ctrl-Z to exit global configuration mode.

Example:

```
Device(config)# end
```

Create policy map (CLI)

Define and configure a policy map in global configuration mode to classify and act on network traffic.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a policy map by entering the policy map name to enter policy-map configuration mode.

Example:

```
Device(config)# policy-map avc-policy
```

By default, no policy maps are defined.

When a policy map is used, it sets the DSCP to zero for IP packets and sets the CoS to zero for tagged packets.

Note
To delete an existing policy map, use the no policy-map policy-map-name global configuration command.

Step 3 Define a traffic classification to enter policy-map class configuration mode.

Example:

```
Device(config-pmap)# class [class-map-name | class-default]
```

By default, no policy maps or class maps are defined.

If a traffic class has already been defined by using the class-map global configuration command, specify its name as class-map-name in this command.

The class-default traffic class is predefined and can be added to any policy. It is always placed at the end of a policy map. The class-default class includes an implied 'match any,' which ensures that all packets that have not already matched other traffic classes are matched by class-default.

Note
To delete an existing class map, use the no class class-map-name policy-map configuration command.

Step 4 Classify IP traffic by setting a new value in the packet.

Example:

```
Device(config-pmap-c)# set dscp new-dscp 45
```

For **dscp new-dscp** , enter a new DSCP value to assign to the classified traffic. The range is zero to 63.

Step 5 Specify the default class so that you can configure or modify its policy.

Example:

```
Device(config-pmap)# class class-default
```

Step 6 Configure the default DSCP.

Example:

```
Device(config-pmap)# set dscp default
```

Step 7 Return to privileged EXEC mode. Alternatively, press Ctrl-Z to exit global configuration mode.

Example:

```
Device(config-pmap)# end
```

AVC-based selective reanchoring

An AVC-based selective reanchoring is a wireless client mobility mechanism that

- reanchors clients as they roam between one controller to another and prevent depletion of available IP addresses, and
- uses AVC profile-based statistics to determine if a client should be reanchored or deferred.

This is useful when a client is actively running a voice or video application defined in the AVC rules.

The reanchoring process also involves deauthentication of anchored clients. The clients get deauthenticated when they do not transmit traffic for the applications listed in the AVC rules while roaming between controllers.

Restrictions for AVC-based selective reanchoring

- This feature operates only in local mode. FlexConnect and fabric modes are not supported.
- This feature does not operate in guest tunneling scenarios or export anchor scenarios.
- After reanchoring, the old IP address is retained until the IP address lease period ends.

Configure the flow exporter (CLI)

Define and configure a flow exporter to specify where and how NetFlow data is exported from the device.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a flow exporter and enter the flow exporter configuration mode.

Example:

```
Device(config)# flow exporter avc-reanchor
```

Note
You can use this command to modify an existing flow exporter too.

Step 3 Set the exporter as local.

Example:

```
Device(config-flow-exporter)# destination local wlc
```

Configure the flow monitor (CLI)

Set up a flexible NetFlow flow monitor to track traffic statistics.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a flow monitor and enter the flexible NetFlow flow monitor configuration mode.

Example:

```
Device(config)# flow monitor fm_avc
```

Note
You can use this command to modify an existing flow monitor too.

Step 3 Specify the name of an exporter.

Example:

```
Device(config-flow-monitor)# exporter avc-reanchor
```

Step 4 Specify the flow record to use to define the cache.

Example:

```
Device(config-flow-monitor)# record wireless avc basic
```

Step 5 Set the active flow timeout, in seconds.

Example:

```
Device(config-flow-monitor)# cache timeout active 60
```

Step 6 Set the inactive flow timeout, in seconds.

Example:

```
Device(config-flow-monitor)# cache timeout inactive 60
```

Configure the AVC reanchoring profile (CLI)

Configure traffic reanchoring for AVC by creating and updating an AVC reanchoring profile using commands.

Before you begin

- Ensure that you use the AVC-Reanchor-Class class map. All other class-map names are ignored by Selective Reanchoring.
- During boot up, the system checks for the existence of the AVC-Reanchor-Class class map. If it is not found, the system creates default protocols, such as jabber-video, WiFi-calling, and others. If the AVC-Reanchor-Class class map is found, configuration changes are not made. Updates to protocols saved to the startup configuration persist across reboots.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the class map.

Example:

```
Device(config)# class-map AVC-Reanchor-Class
```

Step 3 Instruct the device to match with any of the protocols that pass through it.

Example:

```
Device(config-cmap)# match any
```

Step 4 Specify a match to the application name.

Example:

```
Device(config-cmap)# match protocol jabber-audio
```

You can edit the class-map configuration later to add or remove protocols such as **jabber-video and wifi-calling** if needed.

Configure the wireless WLAN profile policy (CLI)

Follow the procedure given below to configure the WLAN profile policy.

Note Starting with Cisco IOS XE Amsterdam 17.1.1, IPv6 flow monitor is supported on Wave 2 APs. When NBAR runs in the controller and the AP operates in local (central switching) mode, you can attach two flow monitors per direction (input and output) and per IP version (IPv4 and IPv6) in a policy profile. In FlexConnect and fabric modes on Wave 2 APs, when NBAR runs on the AP, only one flow monitor per direction (input and output) and per IP version (IPv4 and IPv6) is supported.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the WLAN policy profile and enters wireless policy configuration mode and disable the policy profile.

Example:

```
Device(config)# wireless profile policy default-policy-profile
Device(config-wireless-policy)# shutdown
```

Step 3 Disable central switching.

Example:

```
Device(config-wireless-policy)# no central switching
```

Step 4 Specify the name of the IPv4 ingress flow monitor.

Example:

```
Device(config-wireless-policy)# ipv4 flow monitor fm_avc input
```

Step 5 Specify the name of the IPv4 egress flow monitor.

Example:

```
Device(config-wireless-policy)# ipv4 flow monitor fm_avc output
```

Step 6 Specify the name of the IPv6 ingress and egress flow monitor.

Example:

```
Device(config-wireless-policy)# ipv6 flow monitor fm_v6_avc input
Device(config-wireless-policy)# ipv6 flow monitor fm_v6_avc output
```

Step 7 Enable the policy profile.

Example:

```
Device(config-wireless-policy)# no shutdown
```

Verify AVC reanchoring

Use the commands to verify the AVC reanchoring configuration:

```
Device# show wireless profile policy detailed avc_reanchor_policy

Policy Profile Name           : avc_reanchor_policy
Description                   :
Status                        : ENABLED
VLAN                          : 1
Wireless management interface VLAN        : 34
!
.
.
.
AVC VISIBILITY                : Enabled
Flow Monitor IPv4
  Flow Monitor Ingress Name   : fm_avc
  Flow Monitor Egress Name    : fm_avc
Flow Monitor IPv6
  Flow Monitor Ingress Name   : Not Configured
  Flow Monitor Egress Name    : Not Configured
NBAR Protocol Discovery       : Disabled
Reanchoring                   : Enabled
Classmap name for Reanchoring
  Reanchoring Classmap Name   : AVC-Reanchor-Class
!
.
.
.
  ------------------------------------------------------

Device# show platform software trace counter tag wstatsd chassis active R0 avc-stats debug

Counter Name Thread ID Counter Value
-----------------------------------------------------------------------------
Reanch_deassociated_clients 28340 1
Reanch_tracked_clients 28340 4
Reanch_deleted_clients 28340 3

Device# show platform software trace counter tag wncd chassis active R0 avc-afc debug

Counter Name Thread ID Counter Value
-----------------------------------------------------------------------------
Reanch_co_ignored_clients 30063 1
Reanch_co_anchored_clients 30063 5
Reanch_co_deauthed_clients 30063 4

Device# show platform software wlavc status wncd
Event history of WNCD DB:

AVC key: [1,wlan_avc,N/A,Reanc,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Reanchoring
Flow-mon-name : N/A
Policy-tag : default-policy-tag
Switching Mode : CENTRAL

Timestamp FSM State Event RC Ctx
------------------------- ------------------- -------------------------- ---- ----
06/12/2018 16:45:30.630342 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:28.822780 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:28.822672 2 :READY 2 :FSM_AFM_UNBIND 0 0
```

```
06/12/2018 16:45:15.172073 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:12.738367 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:12.738261 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:01.162689 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:55.757643 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:55.757542 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:04.468749 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:02.18857 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:02.18717 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:38:20.164304 2 :READY 3 :FSM_AFM_SWEEP 0 2
06/12/2018 16:35:20.163877 2 :READY 1 :FSM_AFM_BIND 0 2
06/12/2018 16:35:18.593257 1 :INIT 1 :FSM_AFM_BIND 0 2
06/12/2018 16:35:18.593152 1 :INIT 24:CREATE_FSM 0 0

AVC key: [1,wlan_avc,fm_avc,v4-In,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Flow monitor IPv4 Ingress
Flow-mon-name : fm_avc
Policy-tag : default-policy-tag
Switching Mode : CENTRAL

Timestamp FSM State Event RC Ctx
------------------------- ------------------- -------------------------- ---- ----
06/12/2018 16:45:30.664772 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:28.822499 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:28.822222 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:15.207605 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:12.738105 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:12.737997 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:01.164225 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:55.757266 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:55.757181 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:04.472778 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:02.15413 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:02.15263 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:38:20.164254 2 :READY 3 :FSM_AFM_SWEEP 0 2
06/12/2018 16:35:20.163209 1 :INIT 1 :FSM_AFM_BIND 0 2
06/12/2018 16:35:20.163189 1 :INIT 24:CREATE_FSM 0 0

AVC key: [1,wlan_avc,fm_avc,v4-Ou,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Flow monitor IPv4 Egress
Flow-mon-name : fm_avc
Policy-tag : default-policy-tag
Switching Mode : CENTRAL

Timestamp FSM State Event RC Ctx
------------------------- ------------------- -------------------------- ---- ----
06/12/2018 16:45:30.630764 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:28.822621 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:28.822574 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:15.172357 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:12.738212 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:12.738167 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:01.164048 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:55.757403 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:55.757361 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:04.472561 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:02.18660 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:02.18588 2 :READY 2 :FSM_AFM_UNBIND 0 0
```

```
06/12/2018 16:38:20.164293 2 :READY 3 :FSM_AFM_SWEEP 0 2
06/12/2018 16:35:20.163799 1 :INIT 1 :FSM_AFM_BIND 0 2
06/12/2018 16:35:20.163773 1 :INIT 24:CREATE_FSM 0 0
```

```
Device# show platform software wlavc status wncmgrd

Event history of WNCMgr DB:

AVC key: [1,wlan_avc,N/A,Reanc,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Reanchoring
Flow-mon-name : N/A
Policy-tag : default-policy-tag
Switching Mode : CENTRAL
Policy-profile : AVC_POL_PYATS

Timestamp FSM State Event RC Ctx
------------------------- ------------------- ------------------------- ---- ----
06/12/2018 16:45:30.629278 3 :WLAN_READY 24:BIND_WNCD 0 0
06/12/2018 16:45:30.629223 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.629179 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.510867 2 :PLUMB_READY 22:BIND_IOSD 0 0
06/12/2018 16:45:30.510411 2 :PLUMB_READY 2 :FSM_WLAN_UP 0 0
06/12/2018 16:45:30.510371 2 :PLUMB_READY 1 :FSM_WLAN_FM_PLUMB 0 0
06/12/2018 16:45:28.886377 2 :PLUMB_READY 20:UNBIND_ACK_IOSD 0 0
!

AVC key: [1,wlan_avc,fm_avc,v4-In,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Flow monitor IPv4 Ingress
Flow-mon-name : fm_avc
Policy-tag : default-policy-tag
Switching Mode : CENTRAL
Policy-profile : AVC_POL_PYATS

Timestamp FSM State Event RC Ctx
------------------------- ------------------- ------------------------- ---- ----
06/12/2018 16:45:30.664032 3 :WLAN_READY 24:BIND_WNCD 0 0
06/12/2018 16:45:30.663958 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.663921 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.511151 2 :PLUMB_READY 22:BIND_IOSD 0 0
06/12/2018 16:45:30.510624 2 :PLUMB_READY 2 :FSM_WLAN_UP 0 0
06/12/2018 16:45:30.510608 2 :PLUMB_READY 1 :FSM_WLAN_FM_PLUMB 0 0
06/12/2018 16:45:28.810867 2 :PLUMB_READY 20:UNBIND_ACK_IOSD 0 0
06/12/2018 16:45:28.807239 4 :READY 25:UNBIND_WNCD 0 0
06/12/2018 16:45:28.807205 4 :READY 23:UNBIND_IOSD 0 0
06/12/2018 16:45:28.806734 4 :READY 3 :FSM_WLAN_DOWN 0 0
!

AVC key: [1,wlan_avc,fm_avc,v4-Ou,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Flow monitor IPv4 Egress
Flow-mon-name : fm_avc
Policy-tag : default-policy-tag
Switching Mode : CENTRAL
Policy-profile : AVC_POL_PYATS

Timestamp FSM State Event RC Ctx
------------------------- ------------------- ------------------------- ---- ----
```

```
06/12/2018 16:45:30.629414 3 :WLAN_READY 24:BIND_WNCD 0 0
06/12/2018 16:45:30.629392 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.629380 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.510954 2 :PLUMB_READY 22:BIND_IOSD 0 0
06/12/2018 16:45:30.510572 2 :PLUMB_READY 2 :FSM_WLAN_UP 0 0
06/12/2018 16:45:30.510532 2 :PLUMB_READY 1 :FSM_WLAN_FM_PLUMB 0 0
06/12/2018 16:45:28.886293 2 :PLUMB_READY 20:UNBIND_ACK_IOSD 0 0
06/12/2018 16:45:28.807844 4 :READY 25:UNBIND_WNCD 0 0
06/12/2018 16:45:28.807795 4 :READY 23:UNBIND_IOSD 0 0
06/12/2018 16:45:28.806990 4 :READY 3 :FSM_WLAN_DOWN 0 0
!
```

AVC-based selective reanchoring

An AVC-based selective reanchoring is a wireless client mobility mechanism that

- reanchors clients as they roam between one controller to another and prevent depletion of available IP addresses, and
- uses AVC profile-based statistics to determine if a client should be reanchored or deferred.

This is useful when a client is actively running a voice or video application defined in the AVC rules.

The reanchoring process also involves deauthentication of anchored clients. The clients get deauthenticated when they do not transmit traffic for the applications listed in the AVC rules while roaming between controllers.

Restrictions for AVC-based selective reanchoring

- This feature operates only in local mode. FlexConnect and fabric modes are not supported.
- This feature does not operate in guest tunneling scenarios or export anchor scenarios.
- After reanchoring, the old IP address is retained until the IP address lease period ends.

Configure the flow exporter (CLI)

Define and configure a flow exporter to specify where and how NetFlow data is exported from the device.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a flow exporter and enter the flow exporter configuration mode.

Example:

```
Device(config)# flow exporter avc-reanchor
```

Note
You can use this command to modify an existing flow exporter too.

Step 3 Set the exporter as local.

Example:

```
Device(config-flow-exporter)# destination local wlc
```

Configure the flow monitor (CLI)

Set up a flexible NetFlow flow monitor to track traffic statistics.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create a flow monitor and enter the flexible NetFlow flow monitor configuration mode.

Example:

```
Device(config)# flow monitor fm_avc
```

Note
You can use this command to modify an existing flow monitor too.

Step 3 Specify the name of an exporter.

Example:

```
Device(config-flow-monitor)# exporter avc-reanchor
```

Step 4 Specify the flow record to use to define the cache.

Example:

```
Device(config-flow-monitor)# record wireless avc basic
```

Step 5 Set the active flow timeout, in seconds.

Example:

```
Device(config-flow-monitor)# cache timeout active 60
```

Step 6 Set the inactive flow timeout, in seconds.

Example:

```
Device(config-flow-monitor)# cache timeout inactive 60
```

Configure the AVC reanchoring profile (CLI)

Configure traffic reanchoring for AVC by creating and updating an AVC reanchoring profile using commands.

Before you begin

- Ensure that you use the AVC-Reanchor-Class class map. All other class-map names are ignored by Selective Reanchoring.
- During boot up, the system checks for the existence of the AVC-Reanchor-Class class map. If it is not found, the system creates default protocols, such as jabber-video, WiFi-calling, and others. If the AVC-Reanchor-Class class map is found, configuration changes are not made. Updates to protocols saved to the startup configuration persist across reboots.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the class map.

Example:

```
Device(config)# class-map AVC-Reanchor-Class
```

Step 3 Instruct the device to match with any of the protocols that pass through it.

Example:

```
Device(config-cmap)# match any
```

Step 4 Specify a match to the application name.

Example:

```
Device(config-cmap)# match protocol jabber-audio
```

You can edit the class-map configuration later to add or remove protocols such as **jabber-video and wifi-calling** if needed.

Configure the wireless WLAN profile policy (CLI)

Follow the procedure given below to configure the WLAN profile policy.

Note Starting with Cisco IOS XE Amsterdam 17.1.1, IPv6 flow monitor is supported on Wave 2 APs. When NBAR runs in the controller and the AP operates in local (central switching) mode, you can attach two flow monitors per direction (input and output) and per IP version (IPv4 and IPv6) in a policy profile. In FlexConnect and fabric modes on Wave 2 APs, when NBAR runs on the AP, only one flow monitor per direction (input and output) and per IP version (IPv4 and IPv6) is supported.

Procedure

Step 1 Enter the global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the WLAN policy profile and enters wireless policy configuration mode and disable the policy profile.

Example:

```
Device(config)# wireless profile policy default-policy-profile
Device(config-wireless-policy)# shutdown
```

Step 3 Disable central switching.

Example:

```
Device(config-wireless-policy)# no central switching
```

Step 4 Specify the name of the IPv4 ingress flow monitor.

Example:

```
Device(config-wireless-policy)# ipv4 flow monitor fm_avc input
```

Step 5 Specify the name of the IPv4 egress flow monitor.

Example:

```
Device(config-wireless-policy)# ipv4 flow monitor fm_avc output
```

Step 6 Specify the name of the IPv6 ingress and egress flow monitor.

Example:

```
Device(config-wireless-policy)# ipv6 flow monitor fm_v6_avc input
Device(config-wireless-policy)# ipv6 flow monitor fm_v6_avc output
```

Step 7 Enable the policy profile.

Example:

```
Device(config-wireless-policy)# no shutdown
```

Verify AVC reanchoring

Use the commands to verify the AVC reanchoring configuration:

```
Device# show wireless profile policy detailed avc_reanchor_policy

Policy Profile Name           : avc_reanchor_policy
Description                   :
Status                        : ENABLED
VLAN                          : 1
Wireless management interface VLAN        : 34
!
.
.
.
AVC VISIBILITY                : Enabled
Flow Monitor IPv4
  Flow Monitor Ingress Name   : fm_avc
  Flow Monitor Egress Name    : fm_avc
Flow Monitor IPv6
  Flow Monitor Ingress Name   : Not Configured
  Flow Monitor Egress Name    : Not Configured
NBAR Protocol Discovery       : Disabled
Reanchoring                   : Enabled
Classmap name for Reanchoring
  Reanchoring Classmap Name   : AVC-Reanchor-Class
!
.
.
.
  ------------------------------------------------------
```

```
Device# show platform software trace counter tag wstatsd chassis active R0 avc-stats debug

Counter Name Thread ID Counter Value
-----------------------------------------------------------------------------
Reanch_deassociated_clients 28340 1
Reanch_tracked_clients 28340 4
Reanch_deleted_clients 28340 3
```

```
Device# show platform software trace counter tag wncd chassis active R0 avc-afc debug

Counter Name Thread ID Counter Value
-----------------------------------------------------------------------------
Reanch_co_ignored_clients 30063 1
Reanch_co_anchored_clients 30063 5
Reanch_co_deauthed_clients 30063 4
```

```
Device# show platform software wlavc status wncd
Event history of WNCD DB:

AVC key: [1,wlan_avc,N/A,Reanc,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Reanchoring
Flow-mon-name : N/A
Policy-tag : default-policy-tag
Switching Mode : CENTRAL

Timestamp FSM State Event RC Ctx
------------------------- ------------------- -------------------------- ---- ----
06/12/2018 16:45:30.630342 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
```

```
06/12/2018 16:45:28.822780 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:28.822672 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:15.172073 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:12.738367 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:12.738261 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:01.162689 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:55.757643 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:55.757542 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:04.468749 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:02.18857 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:02.18717 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:38:20.164304 2 :READY 3 :FSM_AFM_SWEEP 0 2
06/12/2018 16:35:20.163877 2 :READY 1 :FSM_AFM_BIND 0 2
06/12/2018 16:35:18.593257 1 :INIT 1 :FSM_AFM_BIND 0 2
06/12/2018 16:35:18.593152 1 :INIT 24:CREATE_FSM 0 0

AVC key: [1,wlan_avc,fm_avc,v4-In,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Flow monitor IPv4 Ingress
Flow-mon-name : fm_avc
Policy-tag : default-policy-tag
Switching Mode : CENTRAL

Timestamp FSM State Event RC Ctx
------------------------- ------------------- -------------------------- ---- ----
06/12/2018 16:45:30.664772 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:28.822499 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:28.822222 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:15.207605 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:12.738105 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:12.737997 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:01.164225 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:55.757266 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:55.757181 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:04.472778 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:02.15413 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:02.15263 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:38:20.164254 2 :READY 3 :FSM_AFM_SWEEP 0 2
06/12/2018 16:35:20.163209 1 :INIT 1 :FSM_AFM_BIND 0 2
06/12/2018 16:35:20.163189 1 :INIT 24:CREATE_FSM 0 0

AVC key: [1,wlan_avc,fm_avc,v4-Ou,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Flow monitor IPv4 Egress
Flow-mon-name : fm_avc
Policy-tag : default-policy-tag
Switching Mode : CENTRAL

Timestamp FSM State Event RC Ctx
------------------------- ------------------- -------------------------- ---- ----
06/12/2018 16:45:30.630764 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:28.822621 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:28.822574 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:15.172357 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:45:12.738212 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:12.738167 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:45:01.164048 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
06/12/2018 16:44:55.757403 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:55.757361 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:04.472561 3 :ZOMBIE 1 :FSM_AFM_BIND 0 2
```

```
06/12/2018 16:44:02.18660 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:44:02.18588 2 :READY 2 :FSM_AFM_UNBIND 0 0
06/12/2018 16:38:20.164293 2 :READY 3 :FSM_AFM_SWEEP 0 2
06/12/2018 16:35:20.163799 1 :INIT 1 :FSM_AFM_BIND 0 2
06/12/2018 16:35:20.163773 1 :INIT 24:CREATE_FSM 0 0
```

```
Device# show platform software wlavc status wncmgrd

Event history of WNCMgr DB:

AVC key: [1,wlan_avc,N/A,Reanc,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Reanchoring
Flow-mon-name : N/A
Policy-tag : default-policy-tag
Switching Mode : CENTRAL
Policy-profile : AVC_POL_PYATS

Timestamp FSM State Event RC Ctx
------------------------- ------------------- -------------------------- ---- ----
06/12/2018 16:45:30.629278 3 :WLAN_READY 24:BIND_WNCD 0 0
06/12/2018 16:45:30.629223 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.629179 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.510867 2 :PLUMB_READY 22:BIND_IOSD 0 0
06/12/2018 16:45:30.510411 2 :PLUMB_READY 2 :FSM_WLAN_UP 0 0
06/12/2018 16:45:30.510371 2 :PLUMB_READY 1 :FSM_WLAN_FM_PLUMB 0 0
06/12/2018 16:45:28.886377 2 :PLUMB_READY 20:UNBIND_ACK_IOSD 0 0
!

AVC key: [1,wlan_avc,fm_avc,v4-In,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Flow monitor IPv4 Ingress
Flow-mon-name : fm_avc
Policy-tag : default-policy-tag
Switching Mode : CENTRAL
Policy-profile : AVC_POL_PYATS

Timestamp FSM State Event RC Ctx
------------------------- ------------------- -------------------------- ---- ----
06/12/2018 16:45:30.664032 3 :WLAN_READY 24:BIND_WNCD 0 0
06/12/2018 16:45:30.663958 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.663921 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.511151 2 :PLUMB_READY 22:BIND_IOSD 0 0
06/12/2018 16:45:30.510624 2 :PLUMB_READY 2 :FSM_WLAN_UP 0 0
06/12/2018 16:45:30.510608 2 :PLUMB_READY 1 :FSM_WLAN_FM_PLUMB 0 0
06/12/2018 16:45:28.810867 2 :PLUMB_READY 20:UNBIND_ACK_IOSD 0 0
06/12/2018 16:45:28.807239 4 :READY 25:UNBIND_WNCD 0 0
06/12/2018 16:45:28.807205 4 :READY 23:UNBIND_IOSD 0 0
06/12/2018 16:45:28.806734 4 :READY 3 :FSM_WLAN_DOWN 0 0
!

AVC key: [1,wlan_avc,fm_avc,v4-Ou,default-policy-tag]
Current state : READY
Wlan-id : 1
Wlan-name : wlan_avc
Feature type : Flow monitor IPv4 Egress
Flow-mon-name : fm_avc
Policy-tag : default-policy-tag
Switching Mode : CENTRAL
Policy-profile : AVC_POL_PYATS
```

```
Timestamp FSM State Event RC Ctx
------------------------- ------------------- -------------------------- ---- ----
06/12/2018 16:45:30.629414 3 :WLAN_READY 24:BIND_WNCD 0 0
06/12/2018 16:45:30.629392 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.629380 3 :WLAN_READY 4 :FSM_BIND_ACK 0 0
06/12/2018 16:45:30.510954 2 :PLUMB_READY 22:BIND_IOSD 0 0
06/12/2018 16:45:30.510572 2 :PLUMB_READY 2 :FSM_WLAN_UP 0 0
06/12/2018 16:45:30.510532 2 :PLUMB_READY 1 :FSM_WLAN_FM_PLUMB 0 0
06/12/2018 16:45:28.886293 2 :PLUMB_READY 20:UNBIND_ACK_IOSD 0 0
06/12/2018 16:45:28.807844 4 :READY 25:UNBIND_WNCD 0 0
06/12/2018 16:45:28.807795 4 :READY 23:UNBIND_IOSD 0 0
06/12/2018 16:45:28.806990 4 :READY 3 :FSM_WLAN_DOWN 0 0
!
```