Troubleshoot Catalyst 9800 Wireless Controllers

Sudha Katgeri
Technical Leader, CX
@SudhaKatgeri

DGTL-BRKEWN-3013

About me

- Sudha Katgeri
- Technical Leader, Cisco TAC
- Wireless CCIE (#45857)
- Being a mom is my superpower
- Wife-i

Agenda

Chapters

1

- Hardware and Software Architecture
- Life of a Packet

2

- New Config Model
- Deployment Considerations

3

- GUI Troubleshooting Dashboard
- IOS-XE Tracing, Packet Capture & Packet Tracer

4

- Health and KPI Monitoring
- Conclusion

Introduction

- Debugging process is different…
  - Simplified Object model
  - “Store and Process”
  - Always On
  - Trace on Failures
- Improvements in Serviceability
  - Traceability
- Large collaboration between TAC/BU/Customers

Agenda

Chapters

- Hardware and Software Architecture
- Life of a Packet

1

- New Config Model
- Deployment Considerations

2

- GUI Troubleshooting Dashboard
- IOS-XE Tracing, Packet Capture & Packet Tracer

3

- Health and KPI Monitoring
- Conclusion

4

Hardware Architecture

Catalyst 9800 Platforms

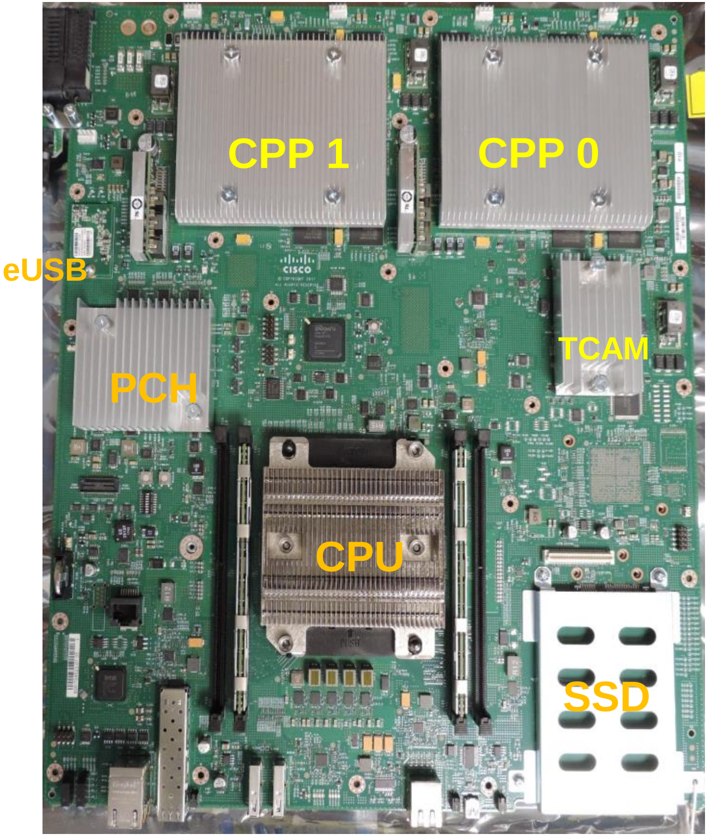
	Embedded Wireless Controller on Access Points (EWC-AP)	**Embedded Wireless Controller on Catalyst 9k Switch (C9800-SW)**	**C9800-L**	**C9800-40**	**9800-80**	**9800-CL (public cloud)**	**9800-CL (private cloud)**
Form Factor	Access Point Form Factor	Switch Form Factor (9300/9400/ 9500 only)	1 RU, ½ width chassis	1 RU appliance	2 RU appliance	AWS, GCP	KVM, Vmware ESXi, Hyper-V, Cisco NFVIS (on ENCS)
Max Supported APs	50/100*	200	250/500**	2,000	6,000	3,000	6,000
Max Supported Clients	1000/2000*	4000	5000/ 10000**	32,000	64,000	32,000	64,000
Deployment Modes	Flexconnect, Mesh	Fabric - SDAccess only (until 17.3) + Central (in 17.3)	Central, Flexconnect, Fabric, Mesh, Flex+bridge	Central, Flexconnect, Fabric, Mesh, Flex+bridge	Central, Flexconnect, Fabric, Mesh, Flex+bridge	Flexconnect only	Central, Flexconnect, Fabric, Mesh

*Only on C9120, C9130 ** Requires valid Performance License

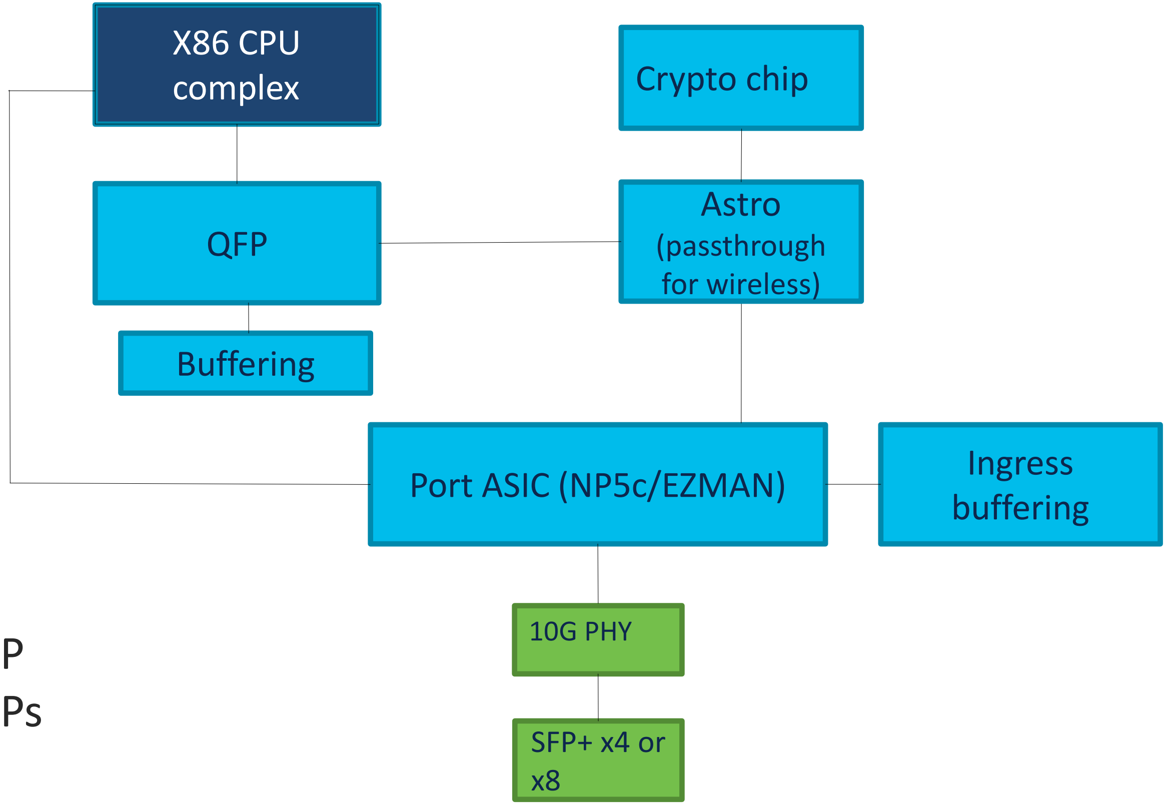
C9800: Cisco Packet Processor (CPP) Data Plane

	9800-40	9800-80	9800-CL
CPP	1 Quantum Flow Processor	2 Quantum Flow Processors (load balanced)	Virtual CPP
CPU	8 cores	12 cores	4/6/10 vCPU
Throughput	40 Gbps	100 Gbps	2.0 Gbps*
Certificate	Manufacturing Installed (MIC)	Manufacturing Installed (MIC)	Self-Signed (SSC)

*For traffic with large (1374 byte) packet size



C9800 Hardware – High Level Block Diagram

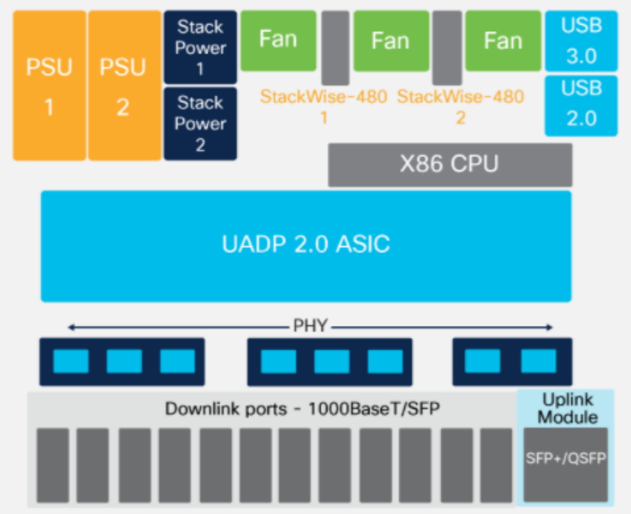


9800-40: 1 QFP
9800-80: 2 QFPs

C9800-SW – Forwarding Engine Driver (FED)/Doppler Data Plane

Outside the scope for this session

- Run on Catalyst 9k Series switches
- Software and Control plane same as CPP platforms
- Cisco Unified Access Data Plane (UADP)
  - Doppler ASIC
  - FED (Forwarding Engine Driver) programs the Doppler
- Controller and Switch accessible via same IP using same CLI and Web UI
- More details in DGTL-BRKARC-2035

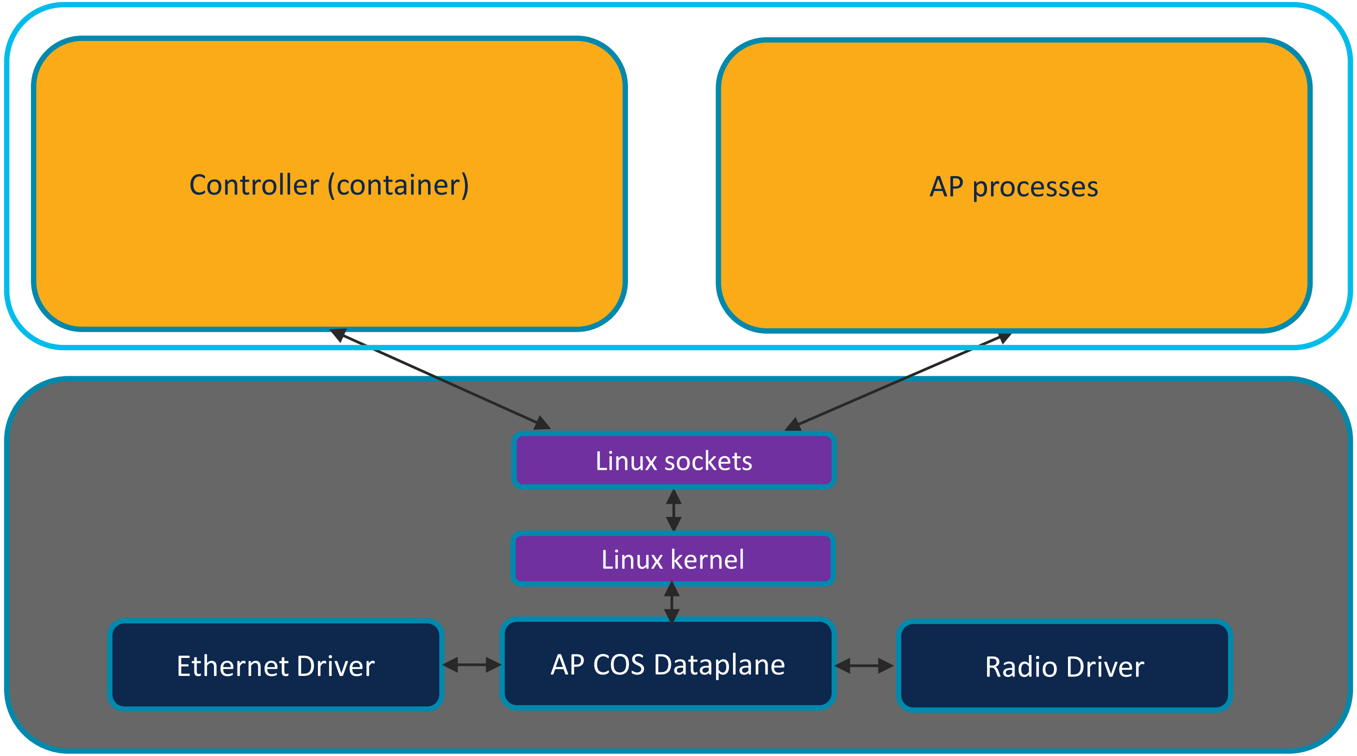


C9800-AP : Embedded Wireless Controller (on AP)

- Only supported on Catalyst 9100 series 11ax APs
  - 9115AX, 9117AX, 9120AX, 9130AX
- Sub-ordinate APs
  - Wave 2 ==18/28/38/4800; 1540/60
  - 11ax== 9115, 9117,9120,9130
- Flash
  - Part 1: AP Primary Image, EWC-AP Image, Config
  - Part 2: AP Backup Image, Logs, Cores, Traces, EWC-AP Image download

EWC architecture

9800 controller on 9100 series APs



TAC Tech Tips – 9800 Appliances

For Your Reference

- Includes field upgradeable components - run latest FPGA/ROMMON
- Uplink ports
  - Configure as trunks
  - Configure **#spanning-tree portfast trunk** on connected switchports
  - Use supported SFPs only – starting 16.12.3 &17.1.1s, link will not come up, if using an unsupported SFP
- Gig1 Service port
  - belongs to Mgmt-Intf vrf by default
  - supports management access via http/https/ssh.
  - Not supported for Netconf telemetry

TAC Tech Tips – 9800 Appliances

For Your Reference

- Wireless Management Interface – use SVI not L3 port.
- Ships with Manufacturing Installed Certificate (MIC) which is used, **by default,** for AP join and mobility tunnel
- There is no need to generate a Self-Signed Certificate.
- Also, do **not** assign any trustpoint to wireless management interface.

TAC Tech Tips – 9800 CL

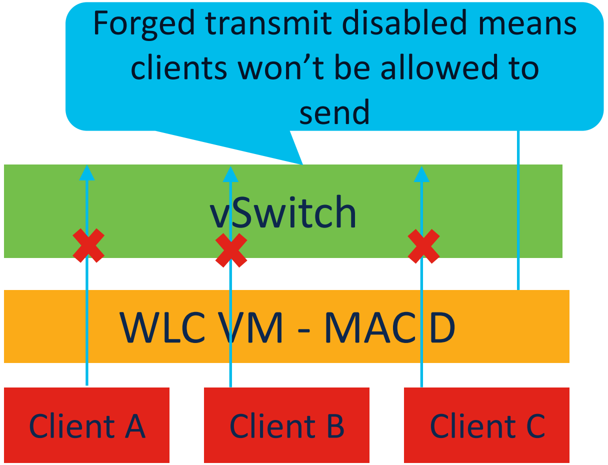
For Your Reference

- Consider the throughput limitation for local mode APs or centrally switched SSIDs
- If deploying on AWS, don’t hope for too much RCA when going unreachable
- 9800-CL shows up with “GigabitEthernet” interfaces but you can set speed to 10000 if the VM NIC supports it
- Needs an SSC for APs to join. SSC can be generated 2 ways
  - Day 0 webUI wizard
  - #wireless config vwlc-ssc key-size 2048 signature-algo sha256 password <yourpassword>
- SSC generation makes use of hostname.

TAC Tech Tips – 9800 CL

For Your Reference

- Forged transmits – Typically disabled for protection against MAC impersonation but needs to be enabled for C9800CL
- Promiscuous mode - has to be enabled on Vmware
- Drawback - All VMs in the same port group and handling the same VLANs will receive each other's traffic. Try to assign WLC VMs to different physical port or different VLANs


Forged transmit disabled means clients won't be allowed to send
vSwitch
WLC VM - MAC D
Client A
Client B
Client C

For Your Reference

TAC Tech Tips – 9800 CL

- Things to keep in mind

- Example of 9800CL on high CPU due to promiscuous mode

```
C9800#show proc cpu platform sorted
CPU utilization for five seconds: 15%, one minute: 15%, five minutes: 16%
Core 0: CPU utilization for five seconds:  3%, one minute:  3%, five minutes:  3%
Core 1: CPU utilization for five seconds:  4%, one minute:  3%, five minutes:  3%
Core 2: CPU utilization for five seconds: 25%, one minute: 18%, five minutes: 19%
Core 3: CPU utilization for five seconds: 29%, one minute: 39%, five minutes: 38%
   Pid    PPid    5Sec    1Min    5Min  Status        Size  Name
--------------------------------------------------------------------------------
 27973   27436     61%     61%     63%  S          222236  ucode_pkt_PPE0
  1030   15026      3%      3%      2%  R         1069784  linux_iosd-imag
   321       2      3%      3%      3%  S               0  ksmd
 30585   30281      0%      0%      0%  S          166852  cli_agent
 30429       1      0%      0%      0%  S            2712  rotee
 30345   29795      0%      0%      0%  S          241800  dbm
 30281   14672      0%      0%      0%  S            4040  pman.sh
 30029       1      0%      0%      0%  S            2640  rotee
```

TAC Tech Tips – 9800 CL

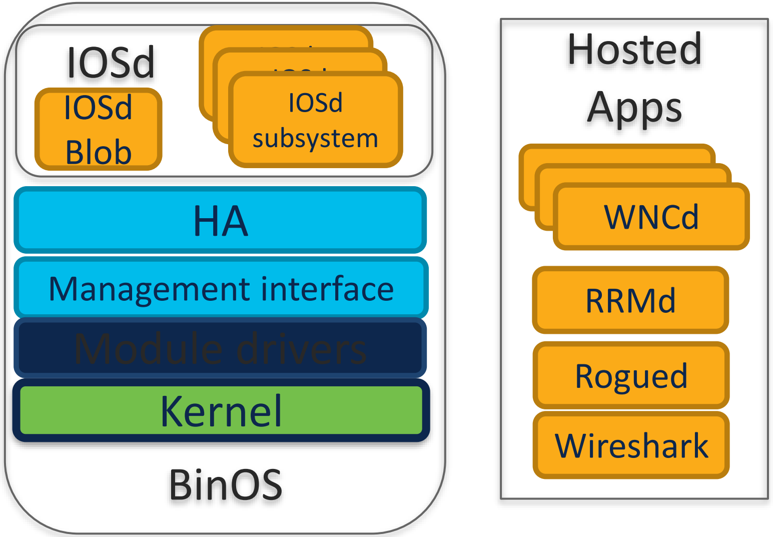
For Your Reference

- VMware ESXi 6.7 and later should have the capability to learn MAC addresses.
- HyperV (IOS-XE 17.1) does not require promiscuous mode
- While bootstrapping, configuring DHCP on Service/Management Interface creates default route off service port and can result in AP Join, client connectivity or traffic forwarding issues.
- Starting 17.3, 9800CL requires 16GB harddisk vs 8GB in previous releases. For existing 9800CL deployments, resizing does not work and 9800CL needs to be redeployed.

Software Architecture

IOS-XE

- Based on BinOS (linux + Cisco patches)
- IOS is now IOSd
- IOS-XE
  - 16.x train – 16.1.x to 16.12.x
  - 17.x train – 17.1.x to 17.3.1
- 16.1 – 16.9 supports switches & routers.
- First IOS-XE for 9800 : Gibraltar 16.10.1
- Since
  - Gibraltar 16.11, 16.12.x
  - Amsterdam 17.1.x, 17.2.x, 17.3.x



Compatibility

- AP Models Supported

11ac wave 1*	1700, 2700, 3700
11ac wave 2	1800, 2800, 3800, 4800, 1540, 1560, 1570
11ax **	9115, 9117, 9120, 9130
IOT APs***	IW3700, IW6300

**11ac wave 1 not supported on EWC-AP*
***Staggered release 9120AXI – 16.12.1; 9120AXE, 9130AXI – 16.12.2; 9130 AXE – 17.1*
***Only APs that can run controller code/EWC-AP*
**** Only supported starting 17.1*

- AP Modes Supported
  - Local, FlexConnect, Monitor, Mesh*, Flex+Mesh*, Sensor, Sniffer

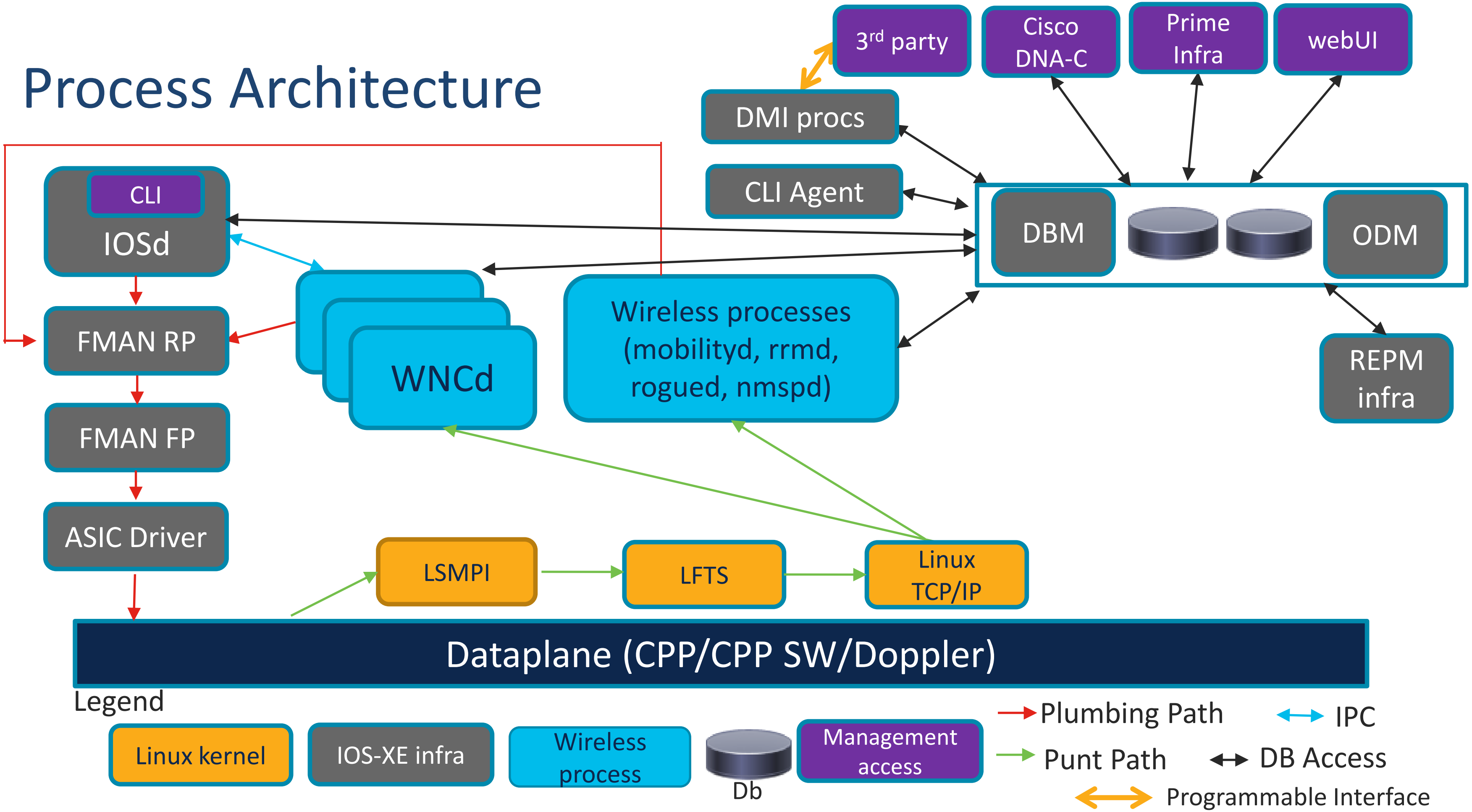
**Only on wave 1 and outdoor wave 2 APs on 16.x. On all APs starting 17.1*

Compatibility

- Follow the compatibility guideline strictly to ensure smooth deployment.
  - Wireless Compatibility Matrix
  - SDA Compatibility Matrix for SDA deployments

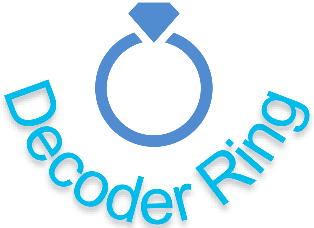
Ex: Prime Infrastructure to C9800 is 1:1 mapping with no backward compatibility.

Process Architecture

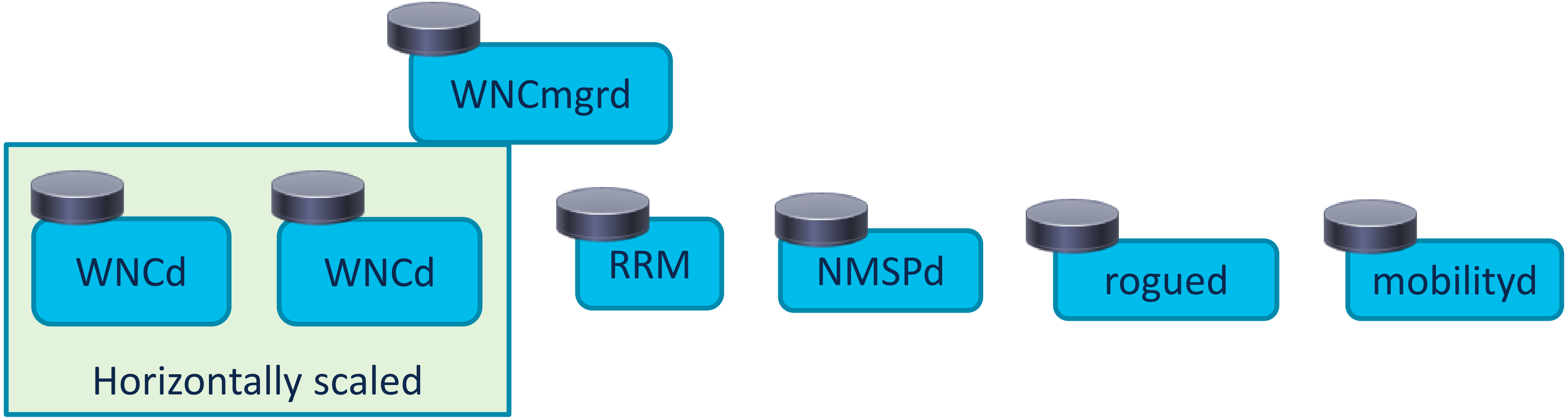


Acronyms

- LSMPI = Linux Shared memory Punt Interface
- LFTS = Linux Forwarding Transport Service
- FMAN = Forwarding Manager
  - FMAN-FP = forwarding processor (Data Plane/DP)
  - FMAN-RP = route processor (Control Plane/CP)
- IOSd = IOS daemon
- DBM = Database Manager
- ODM = Operational Data Manager
- REPM = Replication Manager

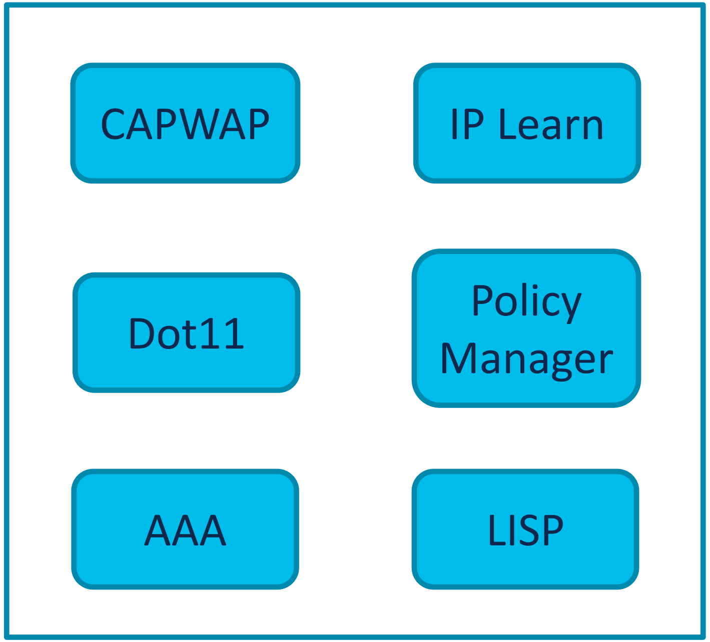


Horizontal Scaling WNCd



WNCd = Wireless Network Control Daemon
RRMd = Radio Resource Manager Daemon
Rogued = Rogue Daemon
NMSPd = NMSP Daemon
Mobilityd = Mobility Daemon

Wireless Network Control Daemon (WNCd)

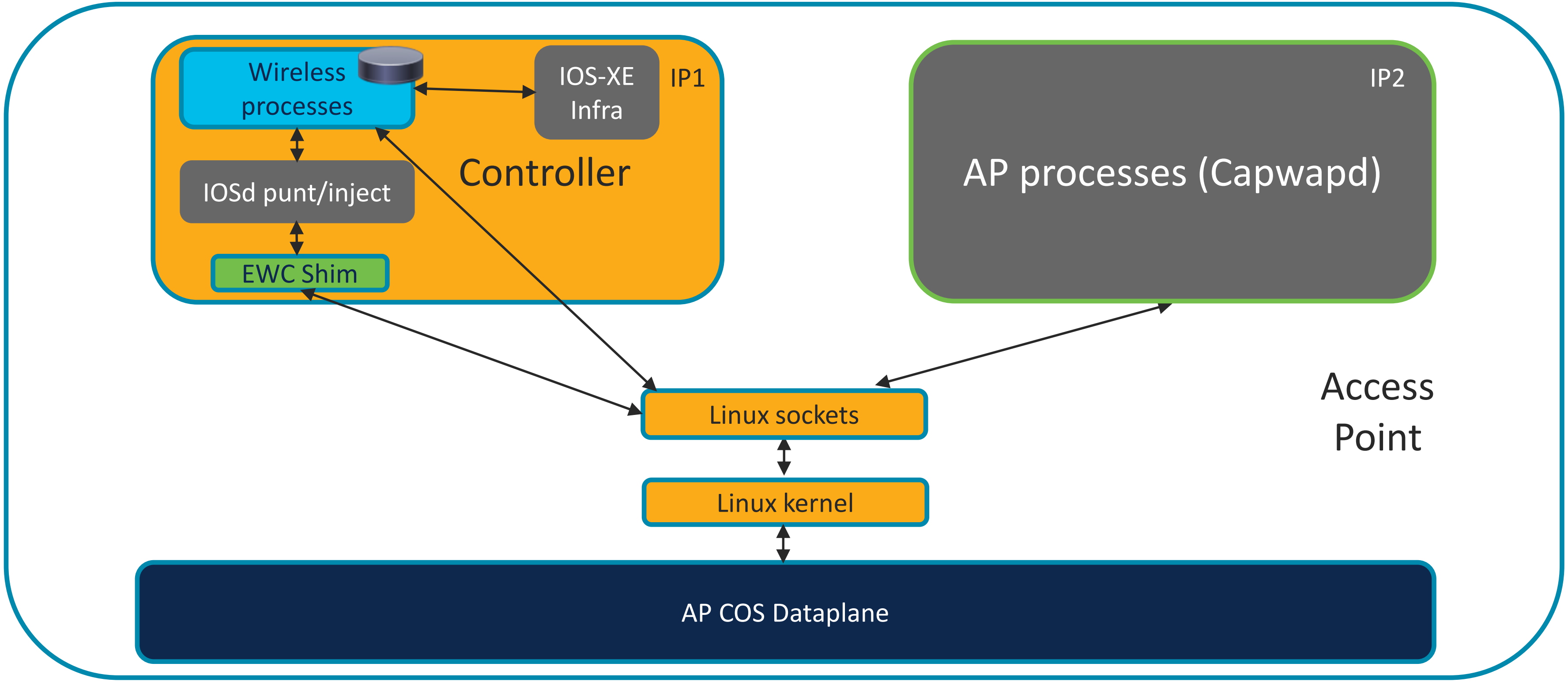


WNCd : controller process managing AP and client session

- Capwap : AP discovery
- Dot11 : Client dot11
- SANET/AAA: Client authentication
- EPM : Client policies
- SISF : client IP learning
- Client Orchestrator : Client State Transitions
- LISP-agent : L2 Lisp handling for Fabric deployment

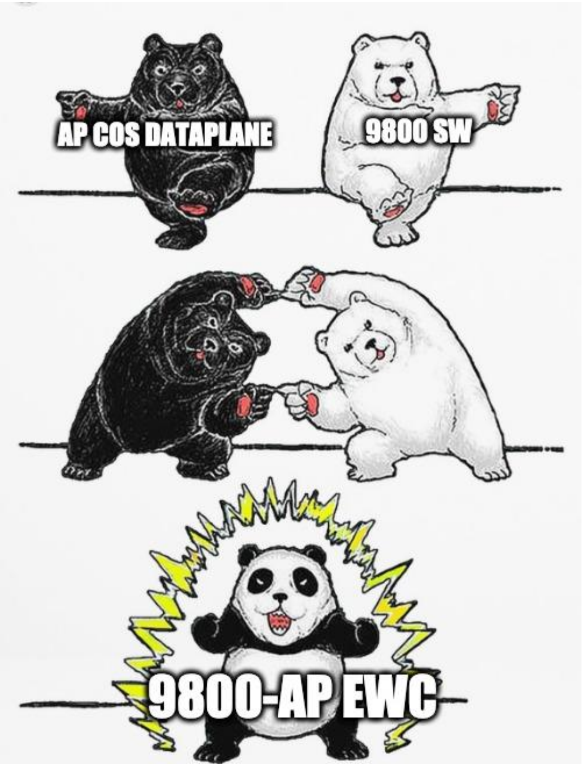
EWC architecture

9800 controller on 9100 series APs



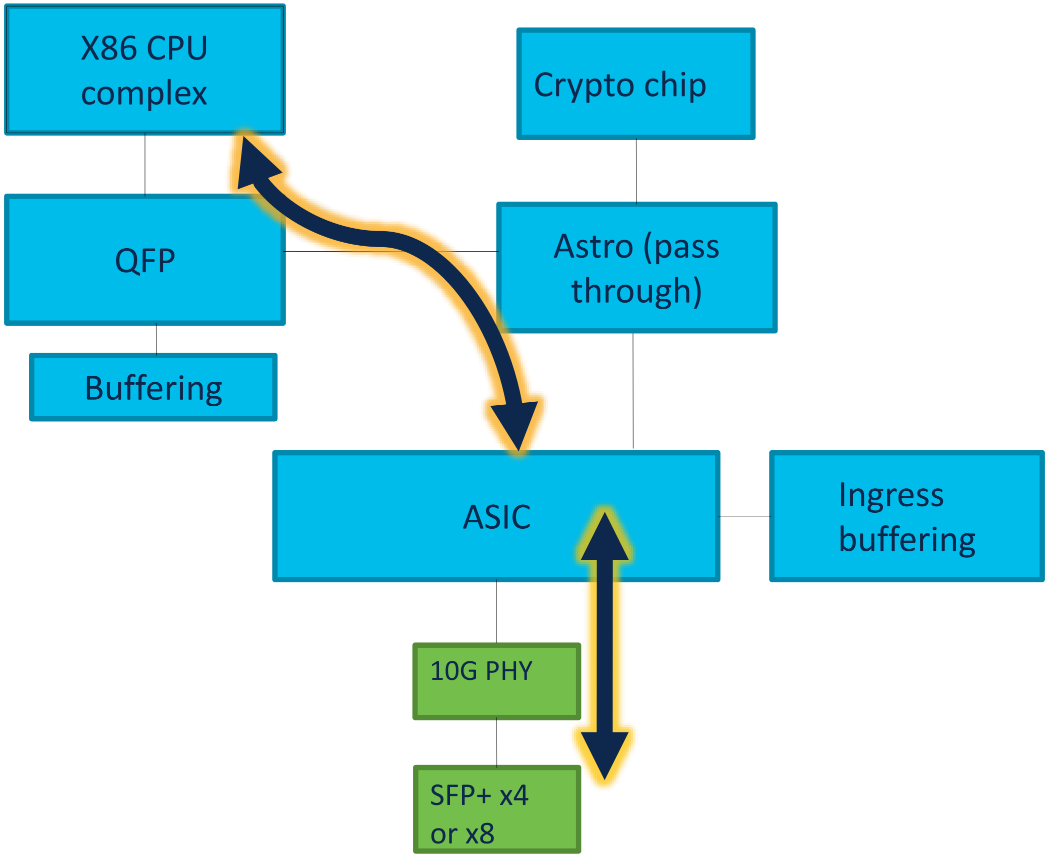
C9800-AP : Embedded Wireless Controller (on AP)

- AP runs linux based AP COS operating system
- Only one AP runs controller code
- Flexconnect only
- EWC software
  - Few processes than c9800
  - Single database
  - Dataplane (AP provides dataplane)

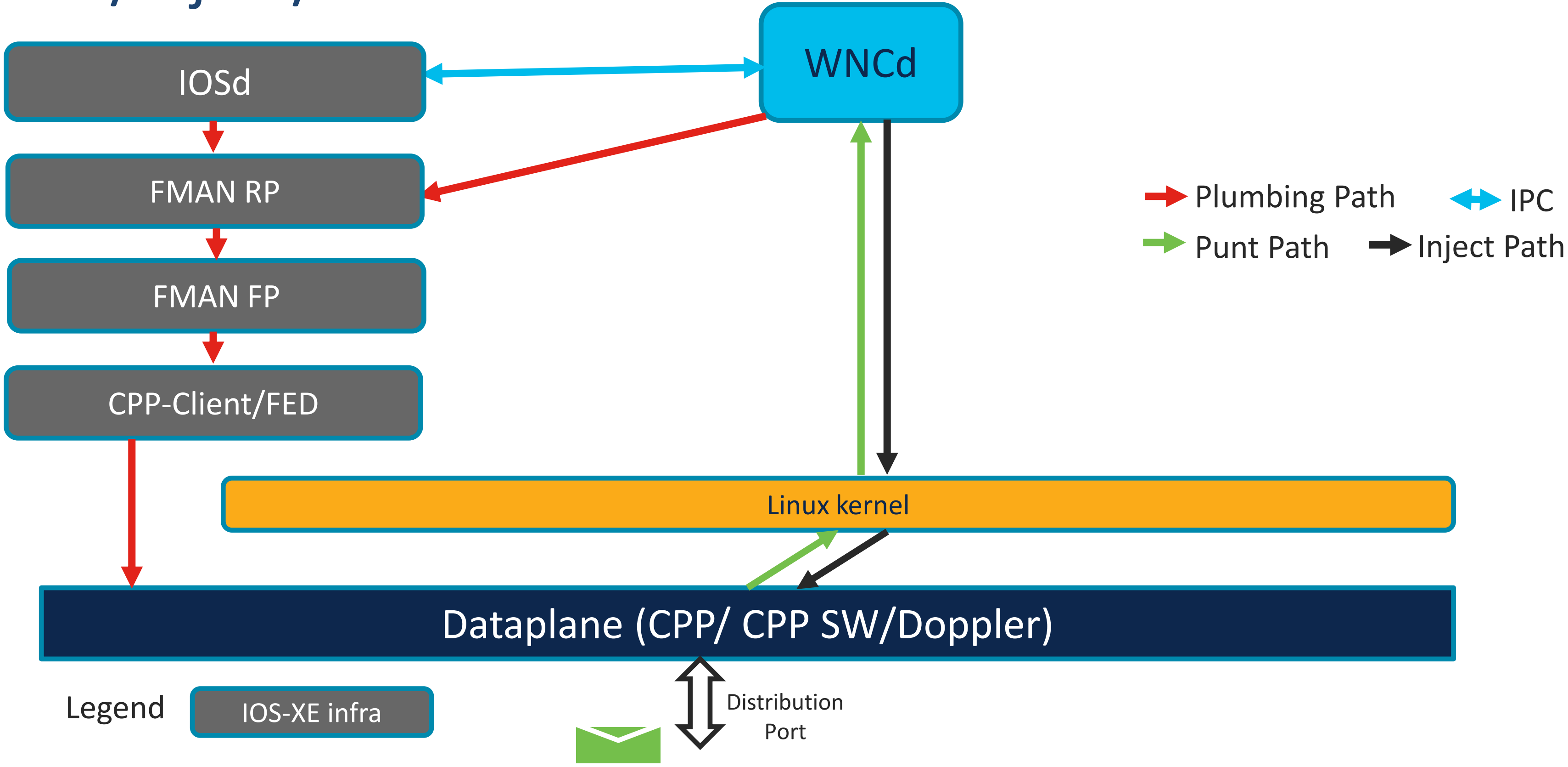


Life of a Packet

Life of a Packet : Control plane

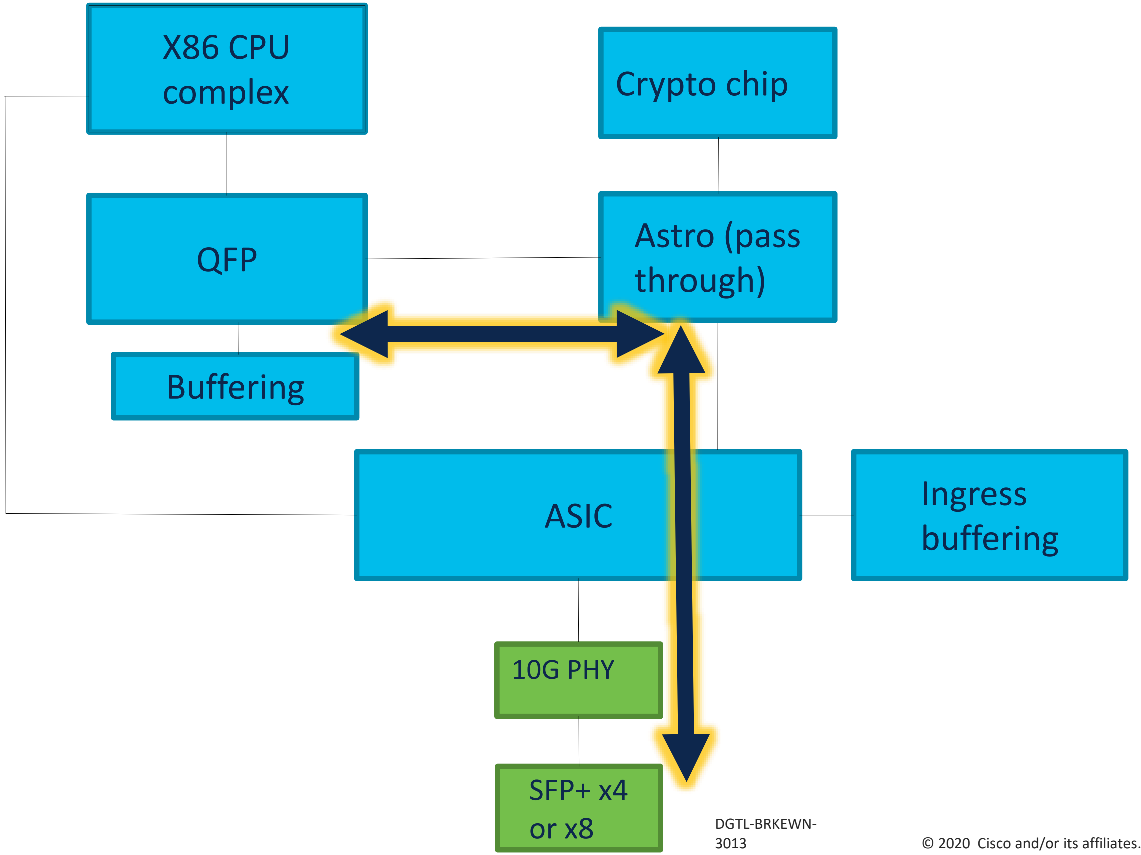


Punt/Inject/Plumb Path



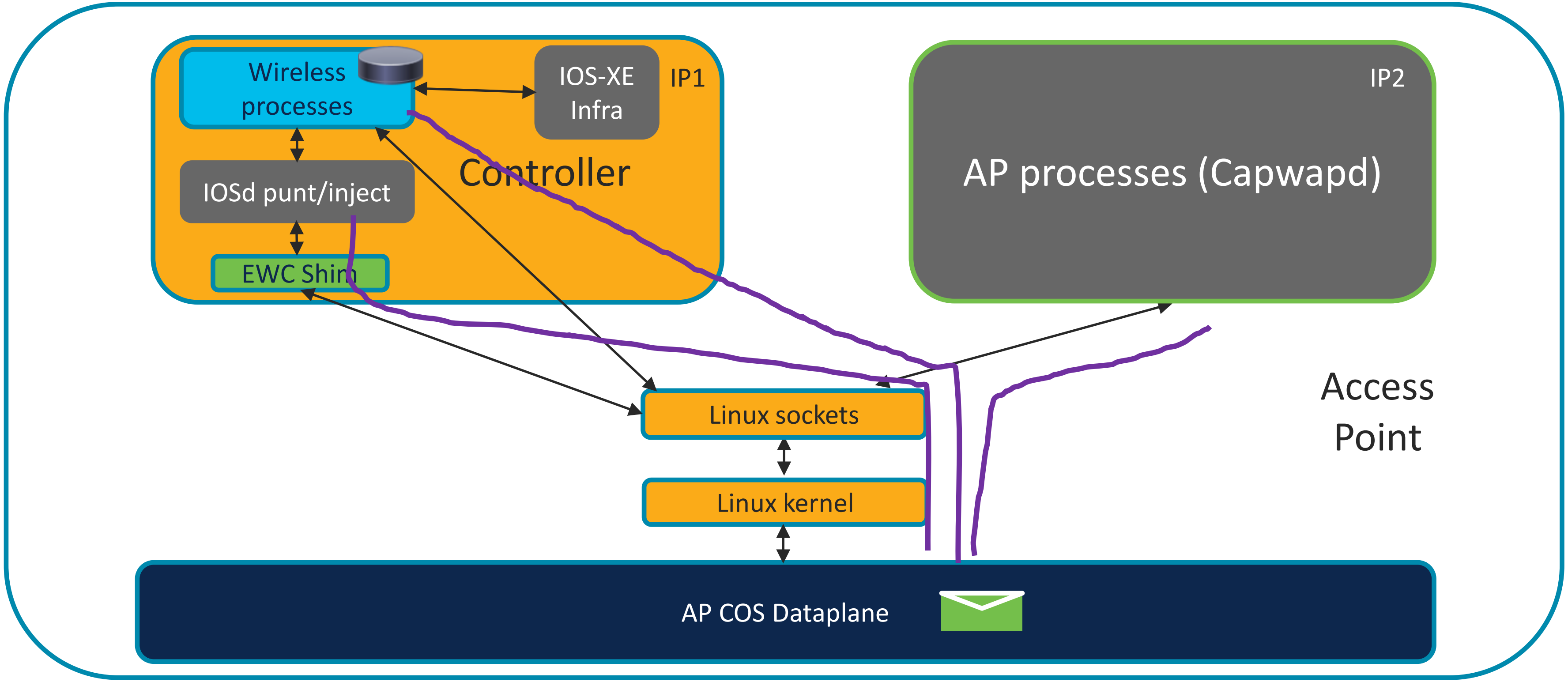
Life of a Packet : Data Plane

wireless client traffic



EWC architecture

9800 controller on 9100 series APs



Agenda

Chapters

- Hardware and Software Architecture
- Life of a Packet

1

- New Config Model
- Deployment Considerations

2

- GUI Troubleshooting Dashboard
- IOS-XE Tracing, Packet Capture & Packet Tracer

3

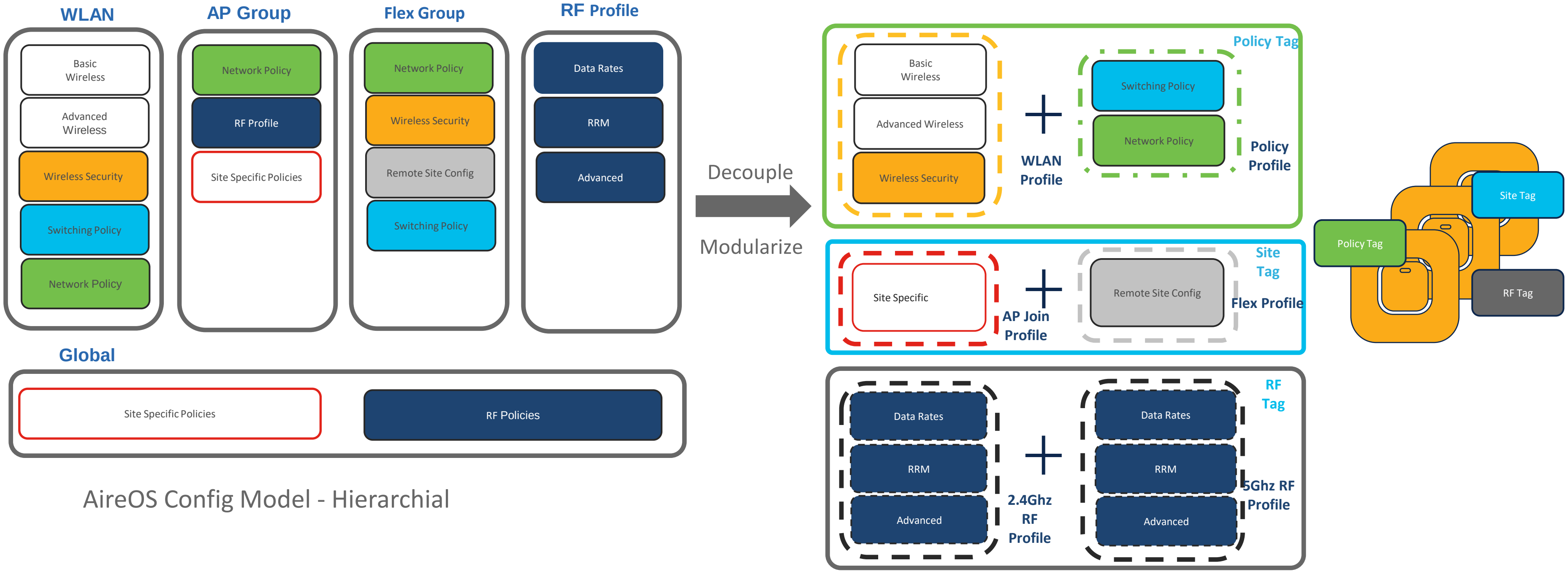
- Health and KPI Monitoring
- Conclusion

4

New Config Model

AireOS vs. Catalyst 9800 Config Model

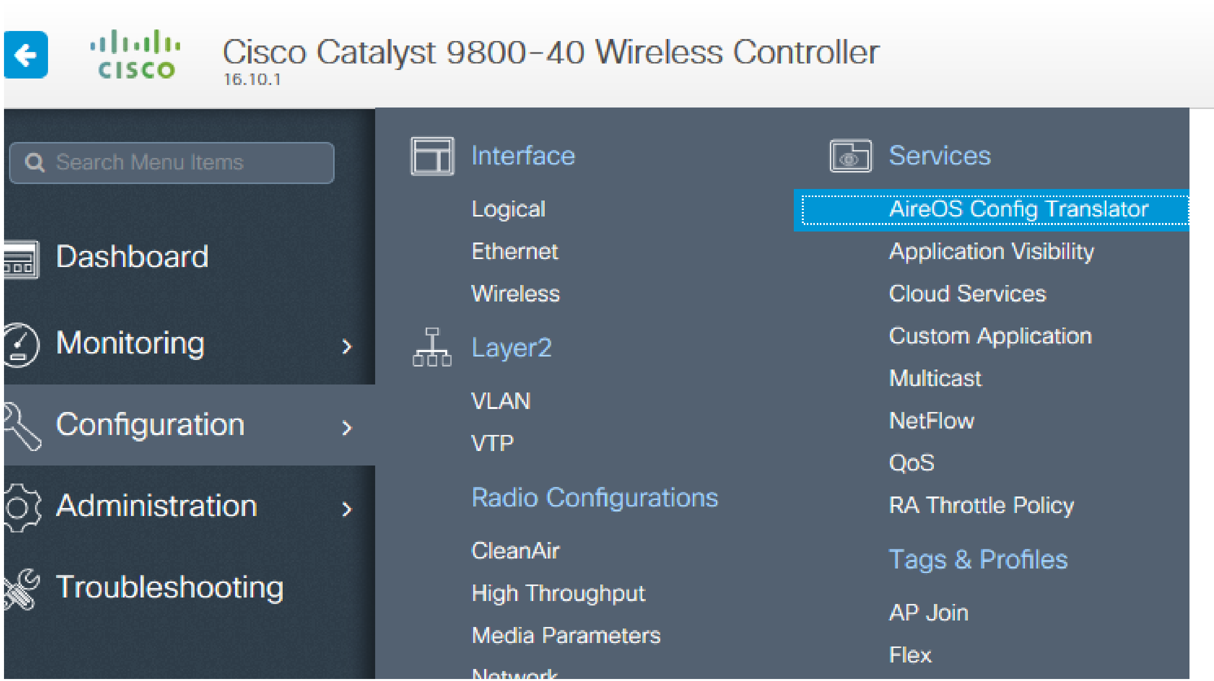
Modularized and Reusable model with **Logical decoupling** of configuration entities



AireOS Config Model - Hierarchial

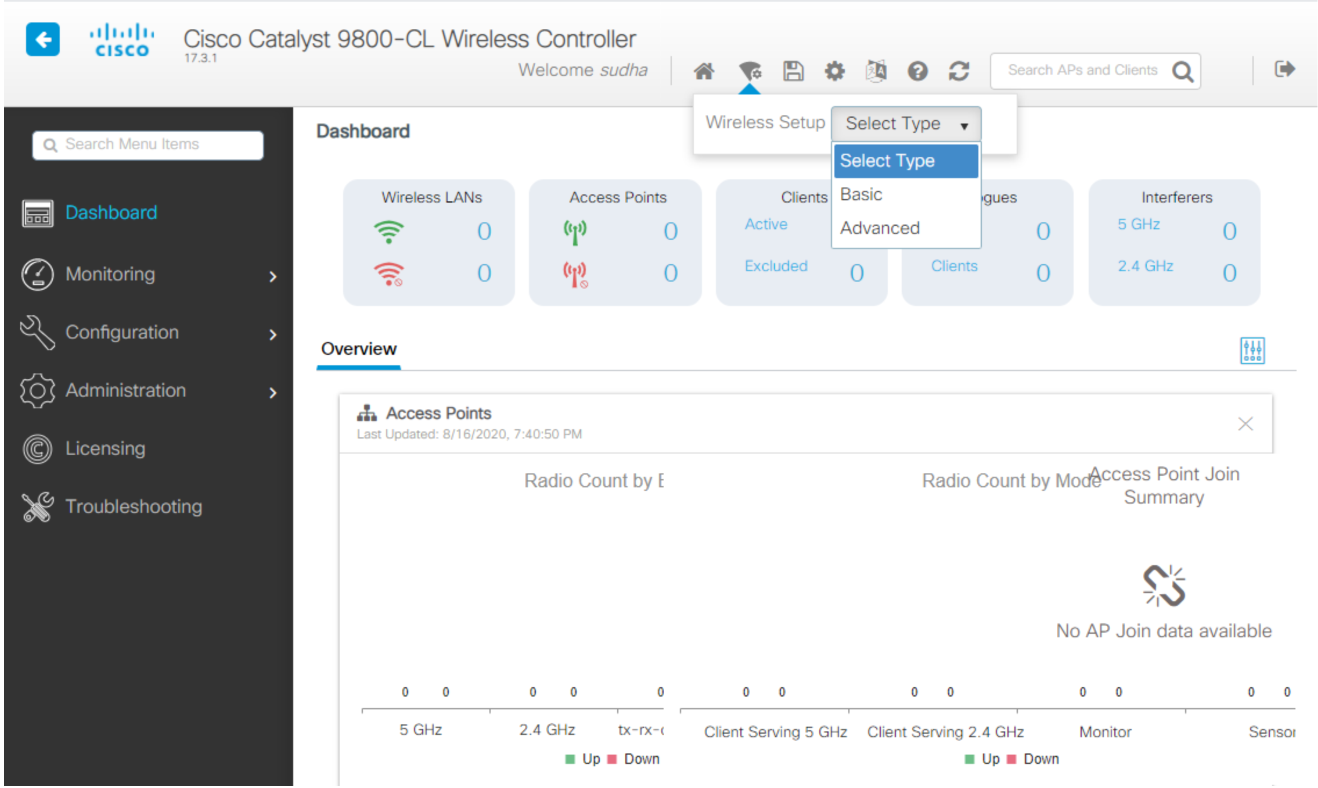
C9800 Config Model – Non-Hierarchial

AireOS to 9800 Configuration Translator



Cloud Tool https://cway.cisco.com/tools/WirelessConfigConverter/

Wireless Setup Wizard

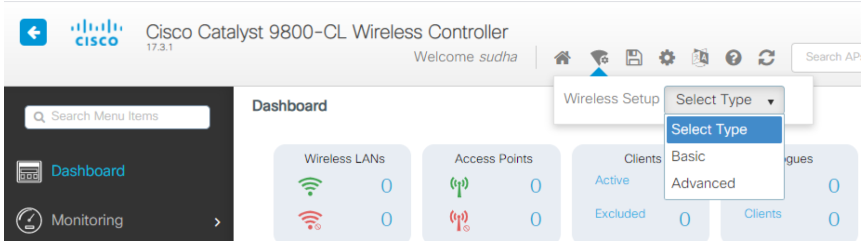


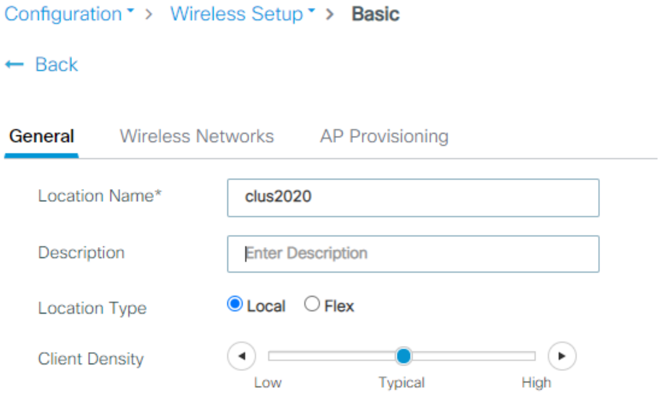
Cisco Catalyst 9800-CL Wireless Controller
17.3.1
Welcome *sudha*

Wireless Setup: Select Type ▾
- Select Type
- Basic
- Advanced

Basic Wireless Setup – TAC Tech Tips

For Your Reference





Location Name* clus2020

Description Enter Description

Location Type ● Local ○ Flex

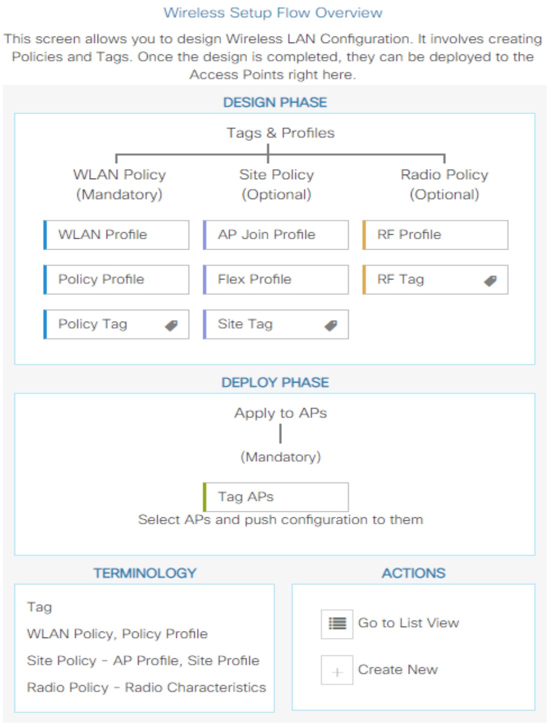
Client Density Low Typical High

- This location is different than AP Location and serves as one of the tag sources for APs.

- Once any advanced configuration, not supported by Basic Wireless Setup, is done on the SSID/policy/RF profiles or corresponding tags; then Basic Setup wizard cannot be re-used to edit (Ex: To modify Location)

Advanced Wireless Setup -

Configuration ˅ > Wireless Setup ˅ > **Advanced**



Wireless Setup Flow Overview

This screen allows you to design Wireless LAN Configuration. It involves creating Policies and Tags. Once the design is completed, they can be deployed to the Access Points right here.

DESIGN PHASE

Tags & Profiles

WLAN Policy (Mandatory)	Site Policy (Optional)	Radio Policy (Optional)
WLAN Profile	AP Join Profile	RF Profile
Policy Profile	Flex Profile	RF Tag
Policy Tag	Site Tag	

DEPLOY PHASE

Apply to APs

(Mandatory)

Tag APs

Select APs and push configuration to them

TERMINOLOGY

Tag

WLAN Policy, Policy Profile

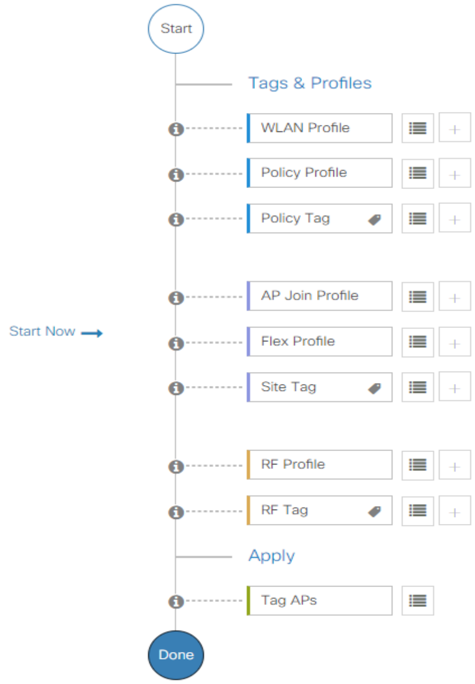
Site Policy - AP Profile, Site Profile

Radio Policy - Radio Characteristics

ACTIONS

Go to List View

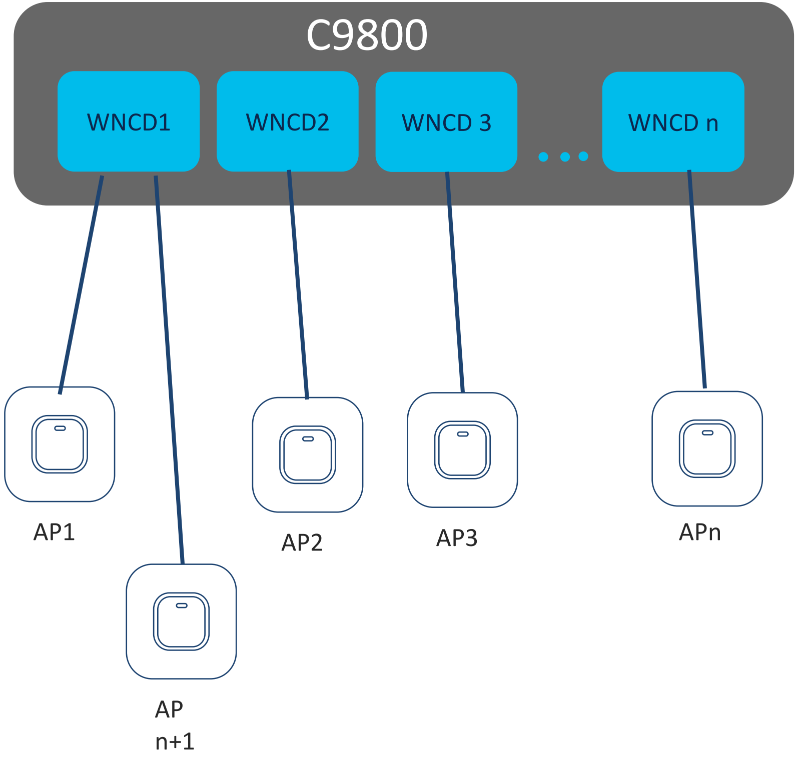
Create New



New Config Model – TAC Tech Tips

For Your Reference

- With no tag config on C9800, AP gets assigned default tags:
  - Default-policy-tag
  - Default-site-tag
  - Default-rf-tag
- APs get loadbalanced across WNCd instances
- Con: Proximity based features like 11k,11v,CHD are managed within each WNCd and will break if neighbors are on different WNCds

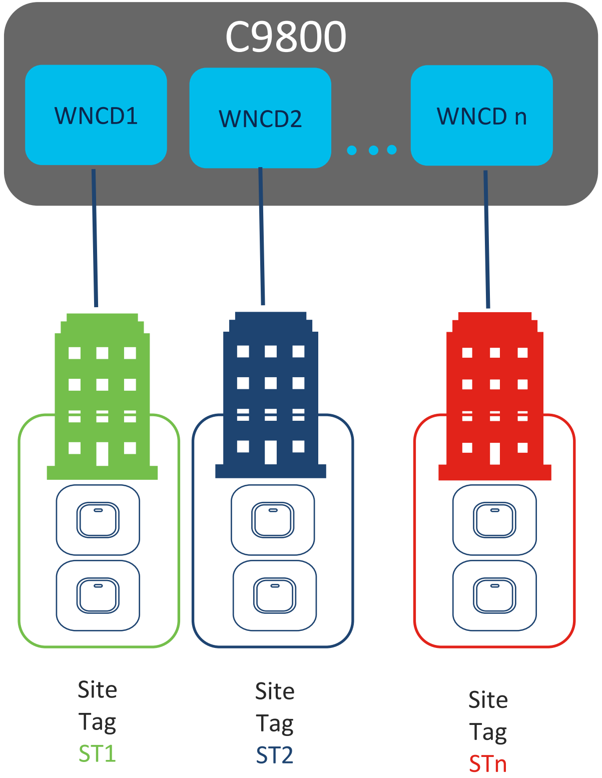


New Config Model – TAC Tech Tips

For Your Reference

- Configure **custom site-tag**
- Assign site-tag based on roaming domain
- For flex, 100 APs per flex site tag
- For local mode AP

	Max APs allowed per site tag	**Max APs recommended per site tag**
9800-40	800	500
9800-80, 9800-CL (med/large)	1600	500

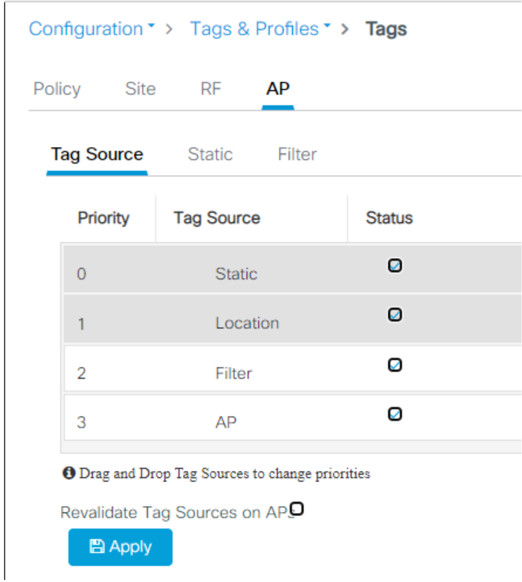


Tag Sources and Priority– TAC Tech Tips

For Your Reference

- Tags are only active after they are applied to one or more APs.
- AP can have multiple tag sources
  - Static – user configured per AP mac
  - Location – Basic Setup Flow
  - Filter – regular expression matching on AP Name
  - AP – tags saved on AP
- These sources are in order of their priority

Statically applied tag is preferred over tags provided by basic setup which, in turn is preferred over filters



Priority	Tag Source	Status
0	Static	☒
1	Location	☒
2	Filter	☒
3	AP	☒

Drag and Drop Tag Sources to change priorities

Revalidate Tag Sources on AP □

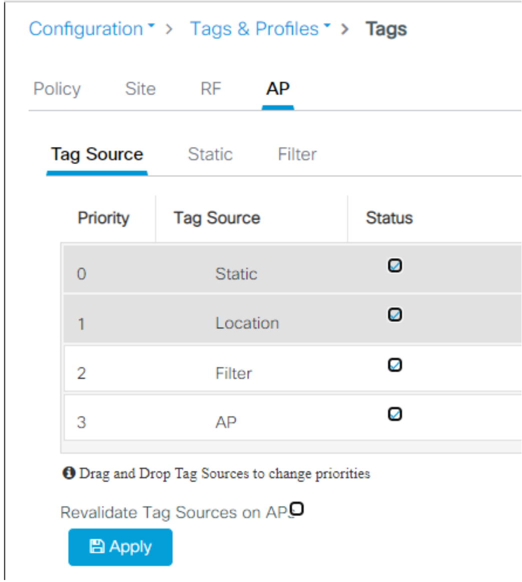
Apply

Tag Sources and Priority – TAC Tech Tips

For Your Reference

- When tags are applied, it does not get saved to the AP persistent memory, by design.
- So, when AP moves to another C9800(say WLC2), it will only inherit tags as per the configuration (static or location or filter) on WLC2 or end up with default tags.
- You can save tags configured to AP nvram: by running

#ap name <APNAME> write tag-config



Priority	Tag Source	Status
0	Static	☑
1	Location	☑
2	Filter	☑
3	AP	☑

Drag and Drop Tag Sources to change priorities

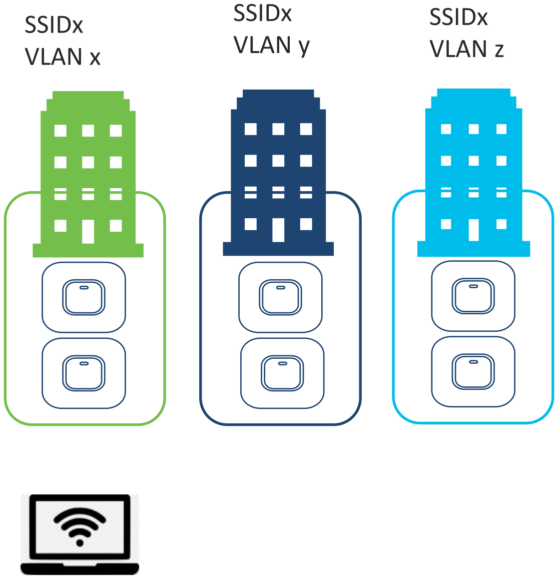
Revalidate Tag Sources on AP □

Roaming across Policy Profiles – TAC Tech Tips

For Your Reference

- Vlan to which wireless clients belong, for a given SSID is defined on the policy profile. Policy tag is then used to map SSID/wlan profile to policy profile.
- On a large campus, multiple policy tags may be in use to map same SSID to different vlans.
- Until 17.3, roaming between APs tagged with different policy profiles was not supported.
- On 17.3, seamless roaming can be achieving by running global config command

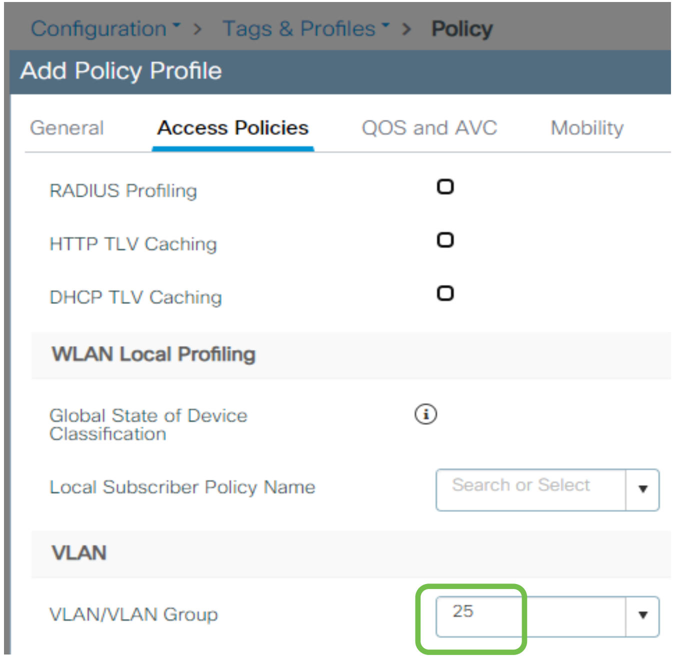
# wireless client-vlan persistent



For Your Reference

FlexConnect Design Philosophy – TAC Tech Tips

- If VLAN ID defined under policy-profile
  - This vlan id dictates the client vlan for flex and no additional vlan mapping is needed under flex profile
  - Vlan id can be native or trunked vlan



RADIUS Profiling □

HTTP TLV Caching □

DHCP TLV Caching □

WLAN Local Profiling

Global State of Device Classification ⓘ

Local Subscriber Policy Name: Search or Select ▾

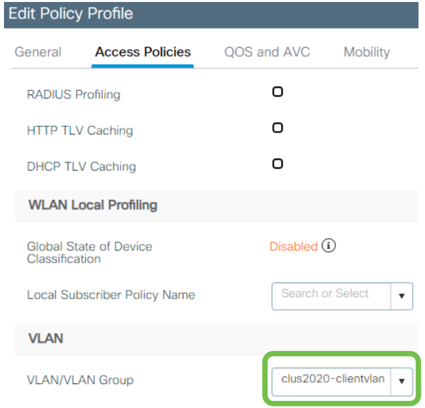
VLAN

VLAN/VLAN Group: 25 ▾

FlexConnect Design Philosophy – TAC Tech Tips

For Your Reference

- If VLAN Name defined under policy-profile
  - Requires **flex** profile to have name to id mapping
  - Same vlan name can be mapped to different ids per flex profile/site tag



Edit Policy Profile

RADIUS Profiling □

HTTP TLV Caching □

DHCP TLV Caching □

WLAN Local Profiling

Global State of Device Classification: Disabled ⓘ

Local Subscriber Policy Name: Search or Select ▾

VLAN

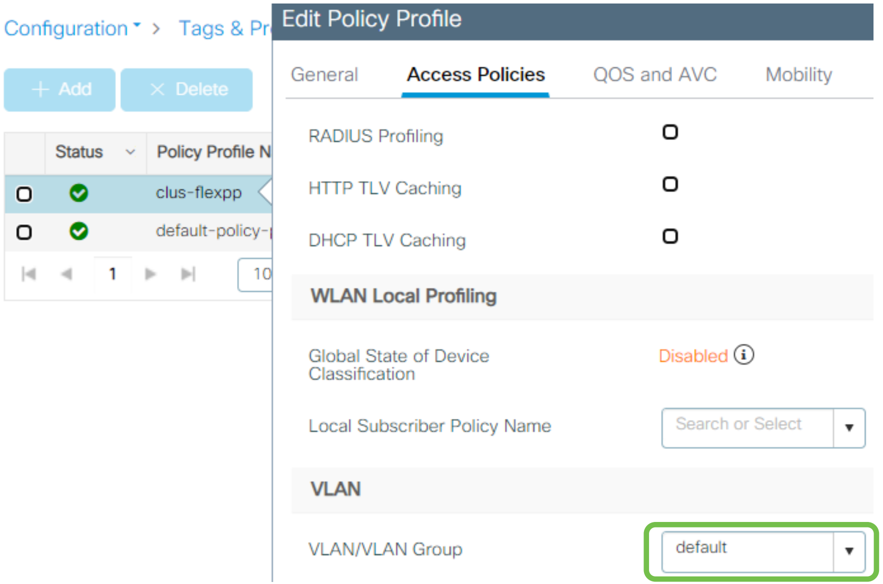
VLAN/VLAN Group: clus2020-clientvlan ▾

Except.....

Vlan 1 vs Vlan-name default (Local Mode) – TAC Tech Tips

For Your Reference

- On c9800, vlan-name default maps to vlan id 1
- If AP is in local mode and client vlan is set to *vlan id* 1 under policy profile, client gets assigned to wireless management vlan (not vlan 1)
    - To assign to vlan 1, use vlan-name default under policy-profile



	Status	Policy Profile N
□	✓	clus-flexpp
□	✓	default-policy-|

RADIUS Profiling □

HTTP TLV Caching □

DHCP TLV Caching □

Global State of Device Classification: Disabled ⓘ

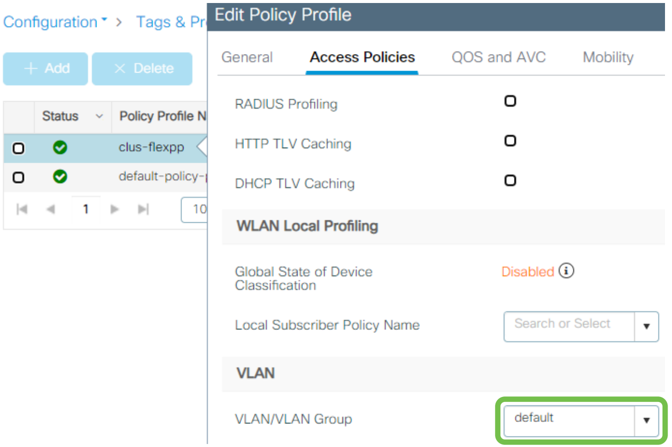
Local Subscriber Policy Name: Search or Select ▾

VLAN/VLAN Group: default ▾

Vlan 1 vs Vlan-name default (Flex) – TAC Tech Tips

For Your Reference

- If AP is in flex mode,
  - if client vlan is set to vlan-id 1 under policy profile, client gets assigned to native vlan for flex AP
  - To assign to vlan 1, use vlan-name default on policy profile. Then map vlan-name default to vlan-id 1 under flex profile.



Configuration > Tags & Pr

+ Add   × Delete

	Status	Policy Profile N
□	✓	clus-flexpp
□	✓	default-policy-

1   10

Edit Policy Profile

General | **Access Policies** | QOS and AVC | Mobility

RADIUS Profiling □

HTTP TLV Caching □

DHCP TLV Caching □

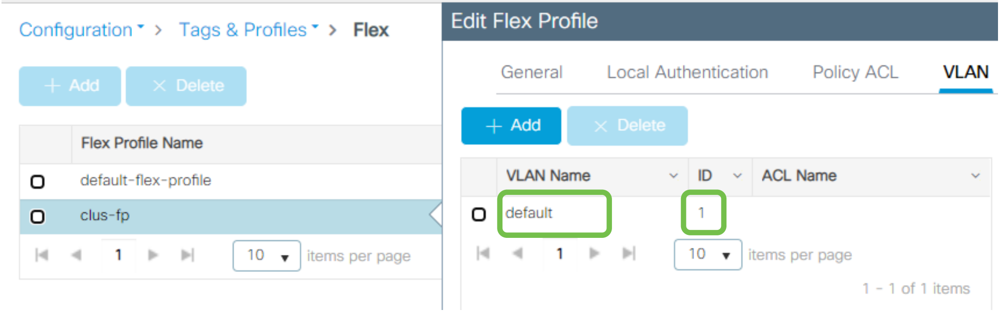
WLAN Local Profiling

Global State of Device Classification: Disabled ⓘ

Local Subscriber Policy Name: Search or Select ▾

VLAN

VLAN/VLAN Group: default ▾



Configuration > Tags & Profiles > Flex

+ Add   × Delete

	Flex Profile Name
□	default-flex-profile
□	clus-fp

1   10 ▾ items per page

Edit Flex Profile

General | Local Authentication | Policy ACL | **VLAN**

+ Add   × Delete

	VLAN Name	ID	ACL Name
□	default	1	

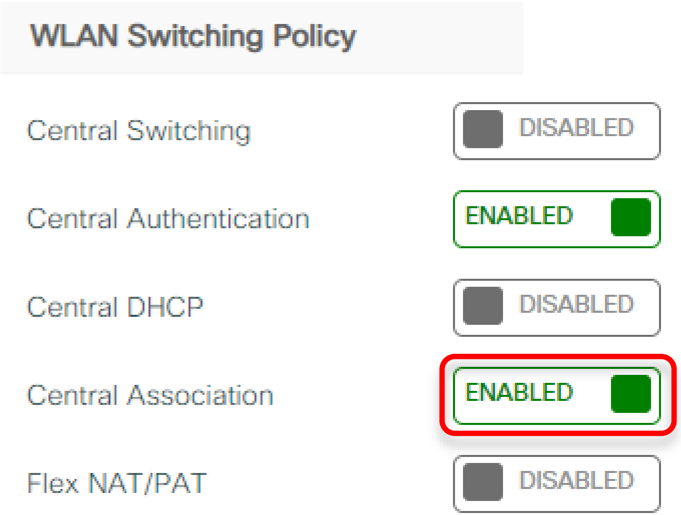
1   10 ▾ items per page

1 – 1 of 1 items

Flex Local Switching/Local Assoc

For Your Reference

- On AireOS, “Flexconnect Central Association” is a niche feature that requires explicit configuration
- On 9800, when policy profile is configured for flex local switching (disabling central switching and central DHCP), it does not automatically disable Central Assoc



WLAN Switching Policy

Central Switching DISABLED

Central Authentication ENABLED

Central DHCP DISABLED

Central Association ENABLED

Flex NAT/PAT DISABLED

Disable Central Assoc for flex policy

Overlapping ip on different Flexconnect sites

- Before 17.3, subnet re-use on different flexconnect sites did not work as 9800 would detect two device with same ip as IP Theft.

- On 17.3, concept of zone was implemented on mac-ip-port binding database to allow for same subnet to exist across different flexconnect sites.

AP tag binding – CLI Verification

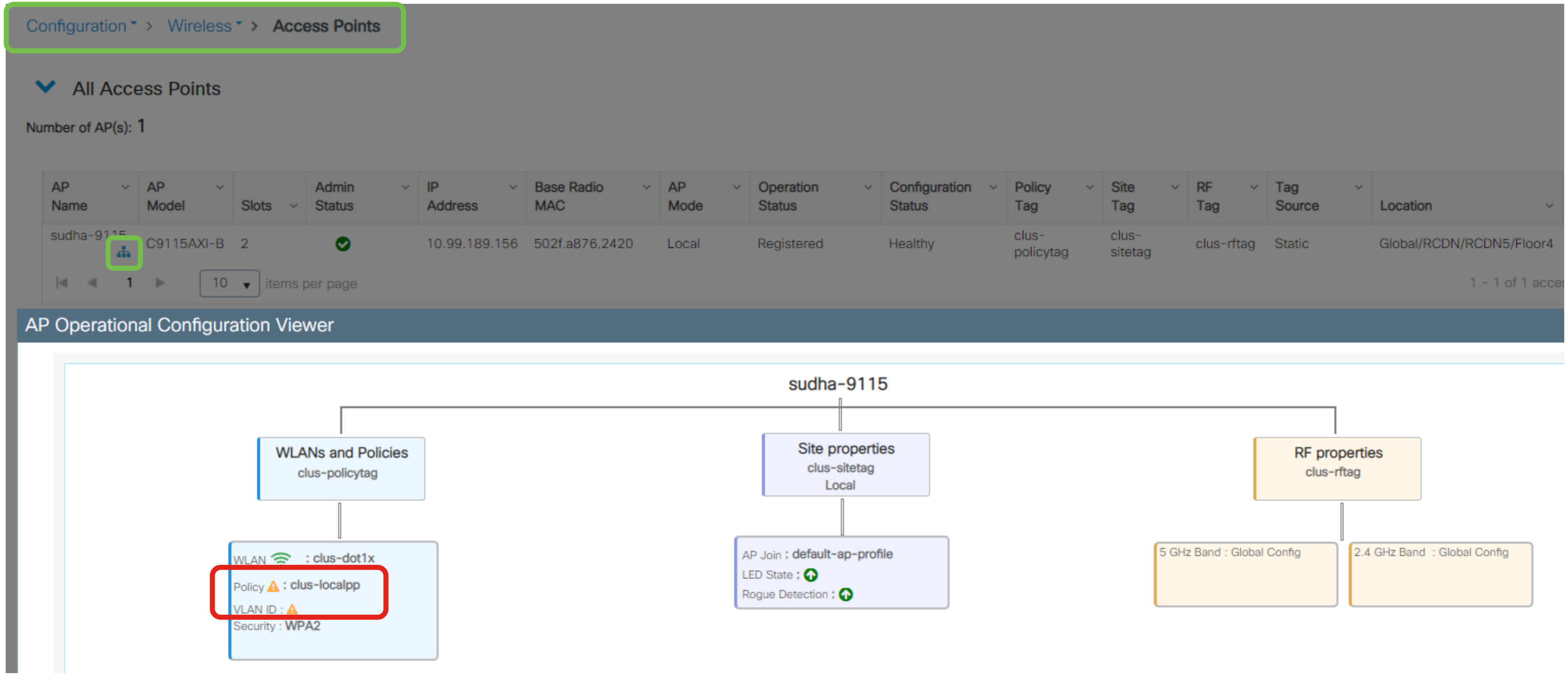
```
9800# show ap tag summary

Number of APs: 1

AP Name        AP Mac              Site Tag Name     Policy Tag Name     RF Tag Name     Misconfigured       Tag Source

---------------------------------------------------------------------------------------------------------

sudha-9115   7069.5a74.8224    sudha-stlocal                   sudha-pt        sudha-rt       No                  Static
```

AP Tag Binding – CLI Verification

```
9800cl-173-1#show ap name sudha-9115 tag detail
AP Name          : sudha-9115
AP Mac          : 7069.5a74.8224

Tag Type          Tag Name
-----------------------------
Policy Tag          clus-policytag
RF Tag             clus-rftag
Site Tag            clus-sitetag

Policy tag mapping
------------------
WLAN Profile Name      Policy Name              VLAN          Flex Central Switching        IPv4 ACL                 IPv6 ACL
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
clus-dot1x               clus-localpp            VLAN1104         ENABLED                     Not Configured               Not Configured

Site tag mapping
----------------
Flex Profile         : default-flex-profile
AP Profile          : default-ap-profile
Local-site          : Yes

RF tag mapping
--------------
5ghz RF Policy       : Global Config
2.4ghz RF Policy     : Global Config
```

New config model

Verifying applied configuration – Web UI



Configuration > Wireless > Access Points

All Access Points

Number of AP(s): 1

AP Name	AP Model	Slots	Admin Status	IP Address	Base Radio MAC	AP Mode	Operation Status	Configuration Status	Policy Tag	Site Tag	RF Tag	Tag Source	Location
sudha-9115	C9115AXI-B	2	✓	10.99.189.156	502f.a876.2420	Local	Registered	Healthy	clus-policytag	clus-sitetag	clus-rftag	Static	Global/RCDN/RCDN5/Floor4

1 10 items per page 1 - 1 of 1 acce

AP Operational Configuration Viewer

sudha-9115

WLANs and Policies
clus-policytag

WLAN : clus-dot1x
Policy : clus-localpp
VLAN ID :
Security : WPA2

Site properties
clus-sitetag
Local

AP Join : default-ap-profile
LED State :
Rogue Detection :

RF properties
clus-rftag

5 GHz Band : Global Config

2.4 GHz Band : Global Config

Configuration Validation

- C9800 has an in-built config validation facility focused on validating profiles and tags configuration.
- You can trigger the config validation by running

```
# wireless config validate
```

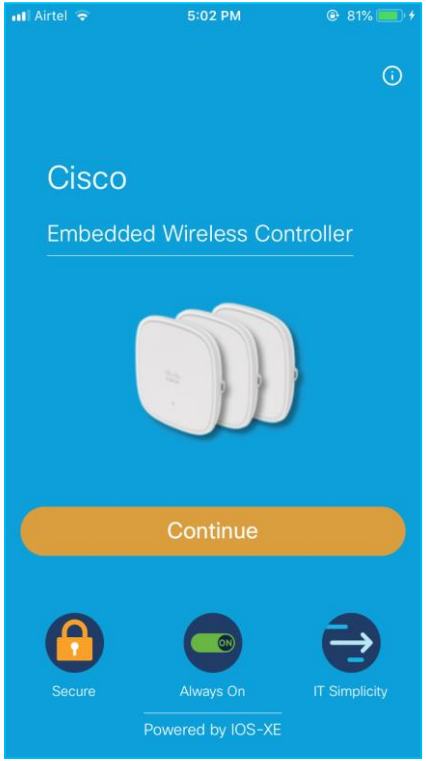
- This will generate a syslog informing of any failures

```
Aug 31 07:44:15.678: %CONFIG_VALIDATOR_MESSAGE-5-EWLC_GEN_ERR: Chassis 1 R0/0: wncmgrd: Error in Policy Tag: clus-policytag; Undefined Element: policy profile, "clus-localpp"
```

- Another tool to view and validate action profiles and tags is Wireless Config Analyzer Express

Deploying EWC-AP, there’s an APP for that !

- Cisco recommends using the Cisco Catalyst Wireless Mobile Application for EWC deployments. The APP is brand new and quite simple to use.
- The mobile application provides the following key benefits:
  - Provision Cisco Embedded Wireless Controller with best practices enabled
  - Monitor real-time performance of the Cisco Embedded Wireless Controller network
  - Manage the Cisco Embedded Wireless Controller network



High Availability

High Availability – Prerequisites

- Platform details must match
  - Same HW model
  - For 9800-CL: Number of cores, memory, storage size
  - Image Version
  - Installation Mode (bundle vs install)
- A mismatch in any of the above results in HA failing to form with a Version Mismatch
- Also note, that VM snapshots are not supported in HA and could lead to failover or crash.

High Availability – V-Mismatch

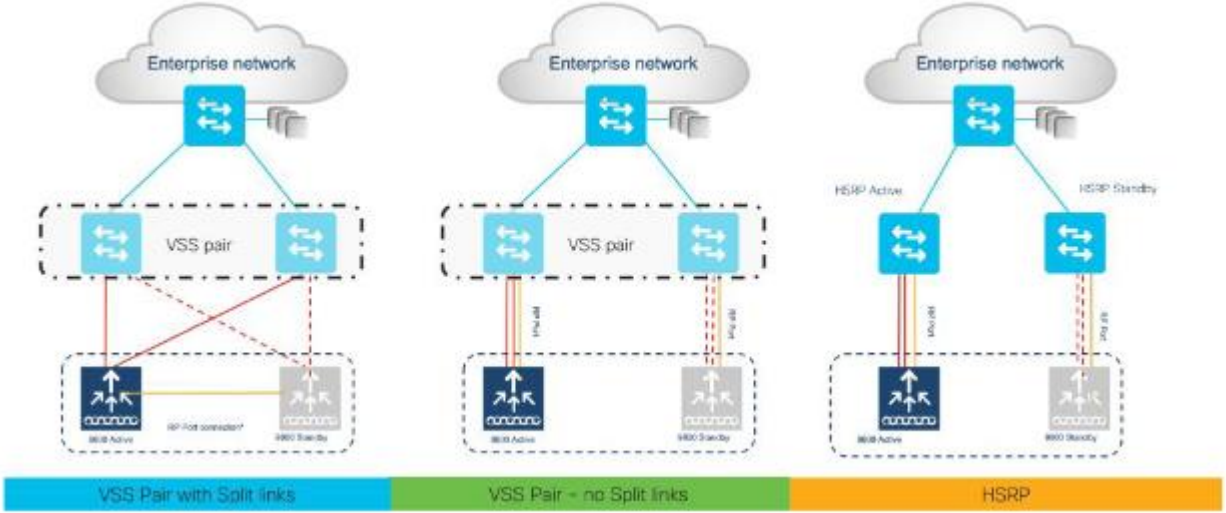
- Pairing between boxes in Install Mode and Bundle mode returns Version Mismatch

```
%BOOT-3-BOOTTIME_INCOMPATIBLE_SW_DETECTED: R0/0: issu_stack: Incompatible software
detected. Details: Active's super boot mode does not match with member's
subpackage boot mode. Please boot switch 1 in super mode.
```

```
C9800-2#sh chassis
Chassis/Stack Mac Address : 00a3.8e23.a0e0 - Local Mac Address
Mac persistency wait time: Indefinite
Local Redundancy Port Type: Twisted Pair
                                             H/W   Current
Chassis#   Role    Mac Address     Priority Version  State                 IP
-------------------------------------------------------------------------------------
 1       Member   00a3.8e23.a320     1      V02     V-Mismatch           192.168.1.171
*2       Active   00a3.8e23.a0e0     1      V02     Ready                192.168.1.172
```

High Availability Supported Deployments - 16.x

- On 16.x releases, there is no gateway reachability check



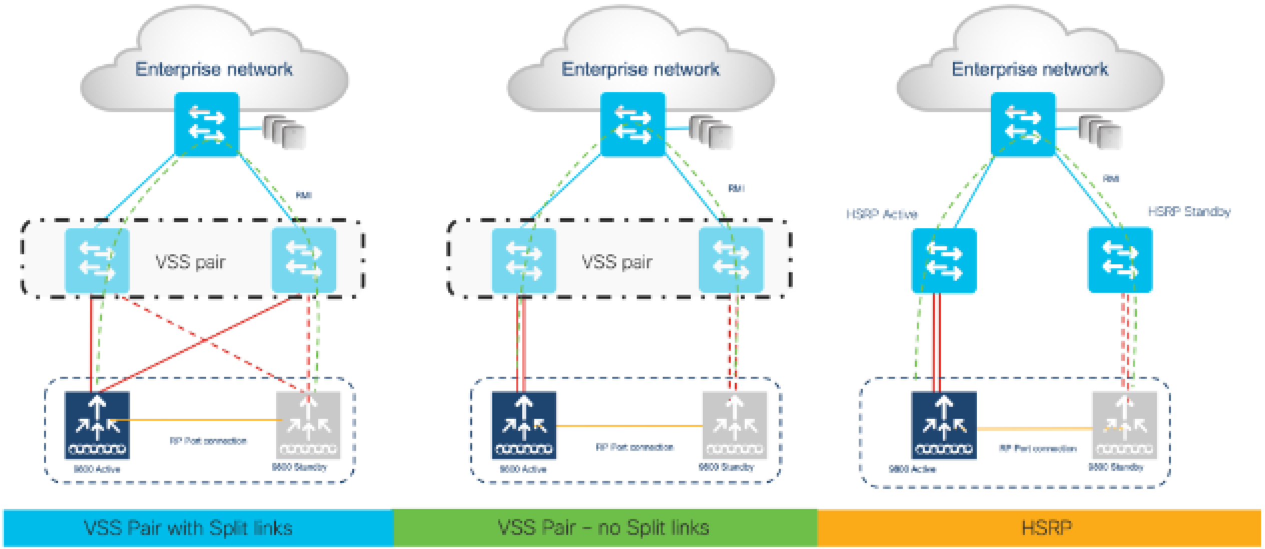
High Availability 16.x – Split Brain Recovery

- If HA sync fails/Split Brain, to recover:
  - Re-ip the boxes to avoid duplicate
  - Regenerate certificates and keys post HA breakup
  - Bounce the http service to get GUI Access

High Availability 17.x Features Added

- Gateway reachability check
- Redundancy management interface
- LACP with HA
- Multi-Lag
- Standby Monitoring without going through Active (17.3)

High Availability supported Deployment – 17.x



High Availability

- General state of HA

```
# show chassis
# show chassis ha-status local
# show chassis ha-status active
# show chassis ha-status standby
# show redundancy
```

- Look back into HA

```
# show redundancy history
# show redundancy switchover history
# show redundancy states
```

High Availability

- Redundancy timers and counters

```
# show platform software stack-mgr chassis active R0 sdp-counters
# show platform software stack-mgr chassis active R0 peer-timeout
# show platform software stack-mgr chassis standby R0 sdp-counters
# show platform software stack-mgr chassis standby R0 peer-timeout
```

- Traces for redundancy

```
# show logging process stack_mgr internal to-file bootflash:<FILENAME.txt>
```

Mobility

Mobility Tunnel Bring-Up

- 9800 supports Secure DTLS Mobility
- Mobility Ports: UDP 16666, 16667
- For mixed deployment (AireOS and 9800), secure mobility needs to be enabled on AireOS side explicitly
- For C9800-CL, SSC hash key needs to be provided to AireOS WLC
- Data DTLS encryption, needs to be enabled or disabled on both ends
- 9800 has a max of 24 WLCs per mobility group, 72 in total

Mobility - Client Roaming

- Seamless roaming requires
  - WLAN Profile Name and SSID need to match
  - WLAN security settings
    - DHCP Required
    - Peer to Peer Blocking
    - 802.11i
    - Various L2/L3 security schemes

```
{wncd_x_R0-0}{1}: [client-orch-sm] [30764]: (ERR): Security Policy Mismatch,
Local: [ ], Remote: [ DHCP ]

{wncd_x_R0-0}{1}: [client-orch-sm] [30764]: (ERR): MAC: aaaa.bbbb.cccc  Handoff
Deny: Security Policy Mismatch
```

AireOS to C9800 – Client Roaming

- For seamless roaming/Inter Release Controller Mobility (IRCM) support between AireOS WLC and C9800
  - Needs 8.8.111.0 or later on 3504, 5520,8540
  - Needs 8.5.164.0 on 5508, 8510
- Same client vlan on both AireOS and C9800 requires 17.3 on IOS-XE and special image on AireOS
- Roam between AireOS and C9800 is always L3 even if same vlan Is defined on AireOS and C9800. Traffic is anchored over to WLC where client roams from.

Mobility

Troubleshooting

- Show tech wireless mobility
- Radio Active tracing using a WLC IP address
- Set platform software trace mobility (...) all-modules debug

Miscellaneous

Smart Licensing

- CSSM
  - Direct from c9800 or via proxy
  - Licenses shared between HA pair
- On-prem CSSM/Satellite server not supported until 17.3
- Use Smart License Reservation (SLR) where cloud CSSM cannot be used
- With SLR, license reserved per SN/chassis

SLR deployment Guide

Interfaces – TAC Tech Tips

- It is recommended to use bridging on c9800 for client traffic and avoid defining SVIs.
- Some features like mdns proxy require L3 interface
- If SVI is defined, some broadcast (directed broadcast in client subnet, L2 broadcast in client subnet etc) are sent out wireless management interface (WMI) as only WMI has a default route of the box.
- VRFs are not supported !

IP Learn – TAC Tech Tips

- If helper-address is configured on L3 SVI for client vlan, DHCP requests will be relayed sourced from client vlan SVI ip address but in wireless management vlan.
  - Firewalls and switches would fail Unicast Reverse Path forward check and drop the relayed packet
- DHCP proxy (in the wlan profile) has the same effect as ip helper.
- No DHCP snooping

Managing C9800 via Prime Infrastructure and DNACenter at same time

- This is supported as long as only one management station is responsible for configuring the box.
  - One mgmt. device will operate in read-write mode
  - Other mgmt. device will be read-only
- CLI, SNMP credentials need to be read-write and Netconf enabled to complete Inventory
- The burden is on network admin to only provision either via DNACenter or Prime Infrastructure and stick to it to prevent unexpected behavior.

Agenda

Chapters

- Hardware and Software Architecture
- Life of a Packet

1

- New Config Model
- Deployment Considerations

2

- GUI Troubleshooting Dashboard
- IOS-XE Tracing, Packet Capture & Packet Tracer

3

- Health and KPI Monitoring
- Conclusion

4

IOS-XE Tracing

IOS-XE Tracing/Debugging

- IOSd Logging
- Binary Tracing
- Always On Tracing
- Trace-on-Failure Summary
- Conditional Debugging/Radioactive Tracing
- Non-Conditional Debugging/Per Process Tracing

IOS-XE Tracing – BinOS Trace Levels

- **ERROR** level represent abnormal situations. We want to raise the user attention to these
- **WARNING** represent an incident that could potentially lead to an error (or not…)
- **NOTICE** is the default logging level for binos daemons. It captures significant events if they are normal working conditions. (client connect, failover)
- **INFO** contains details about state machines and the communication flow
- **DEBUG** contains traces needed to root cause failure conditions
- **VERBOSE** :

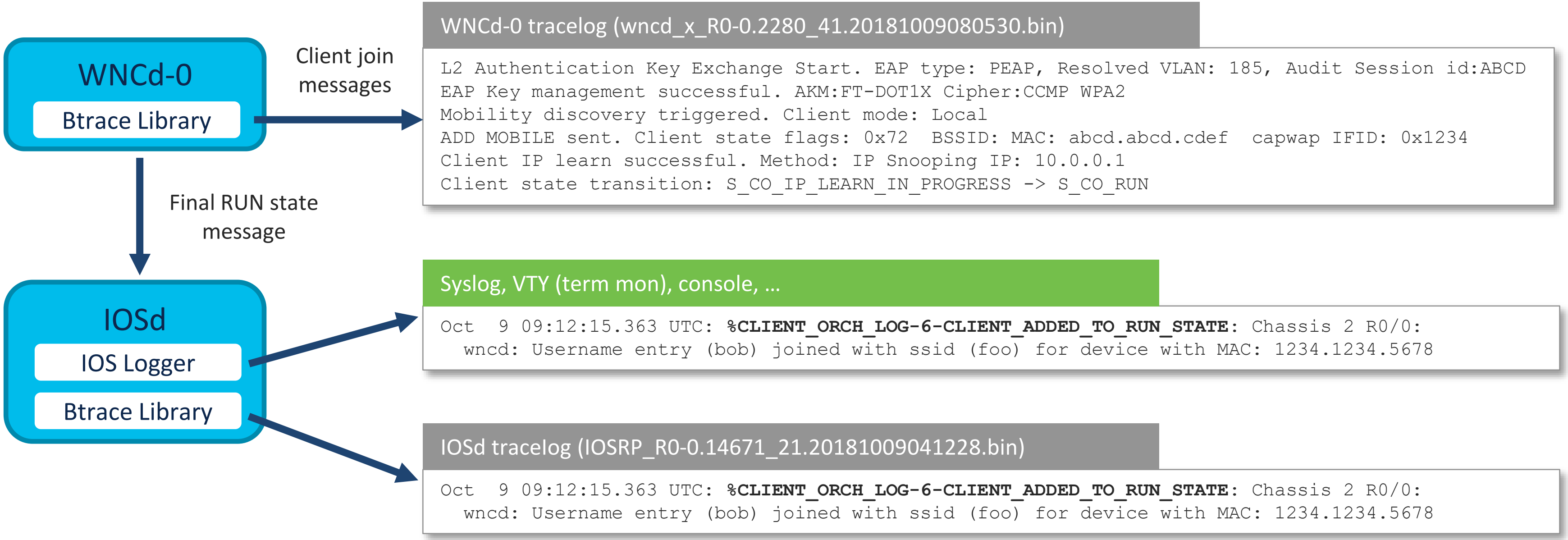
- INTERNAL is not a level but a flag on any log line when it is not meant to be understood by mere mortals but only by developers

2-Critical
3-Error
4-Warning
5-Notice
6-Info
7-Debug
8-Verbose

Syslogs

IOS-XE Logging architecture

IOSd logging Vs btrace



WNCd-0 tracelog (wncd_x_R0-0.2280_41.20181009080530.bin)

```
L2 Authentication Key Exchange Start. EAP type: PEAP, Resolved VLAN: 185, Audit Session id:ABCD
EAP Key management successful. AKM:FT-DOT1X Cipher:CCMP WPA2
Mobility discovery triggered. Client mode: Local
ADD MOBILE sent. Client state flags: 0x72  BSSID: MAC: abcd.abcd.cdef  capwap IFID: 0x1234
Client IP learn successful. Method: IP Snooping IP: 10.0.0.1
Client state transition: S_CO_IP_LEARN_IN_PROGRESS -> S_CO_RUN
```

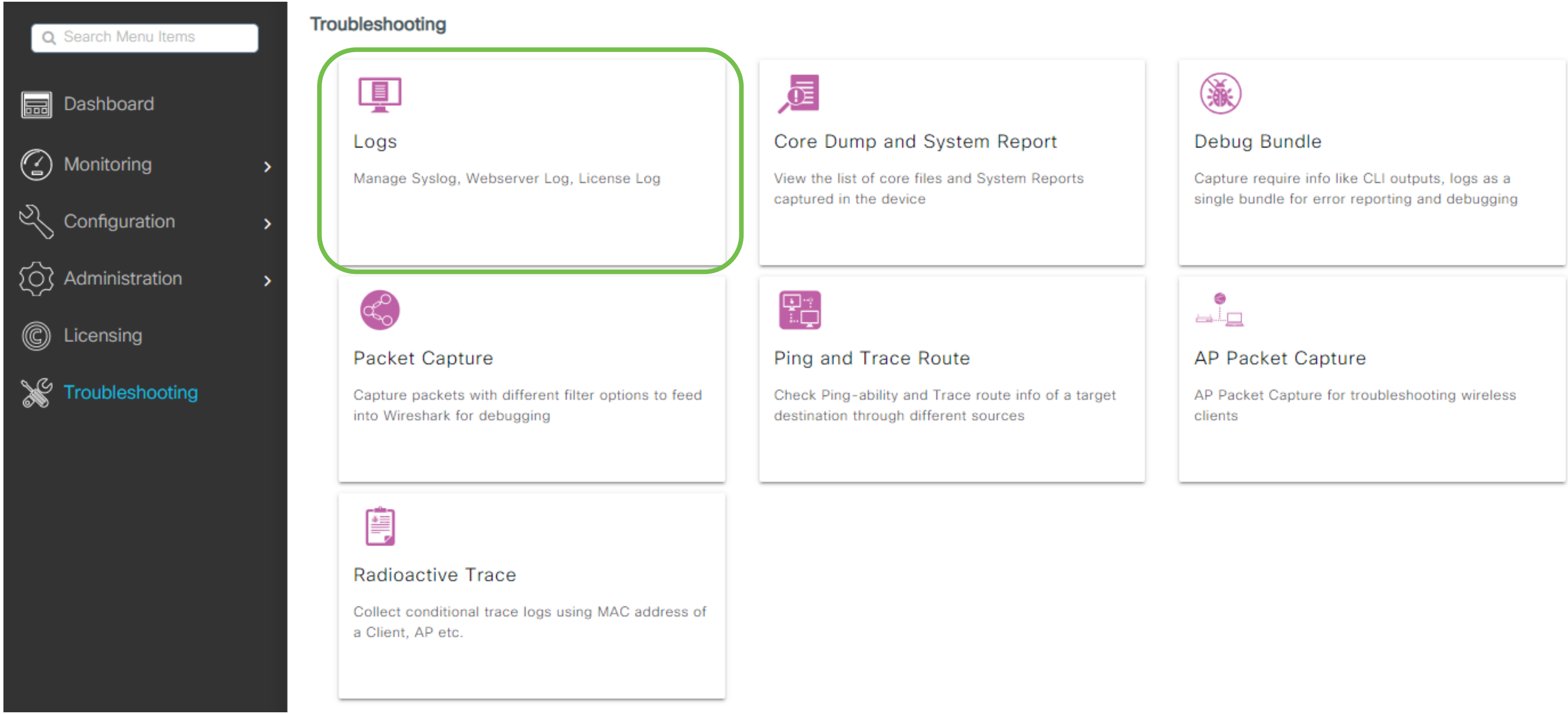
Syslog, VTY (term mon), console, …

```
Oct  9 09:12:15.363 UTC: %CLIENT_ORCH_LOG-6-CLIENT_ADDED_TO_RUN_STATE: Chassis 2 R0/0:
  wncd: Username entry (bob) joined with ssid (foo) for device with MAC: 1234.1234.5678
```

IOSd tracelog (IOSRP_R0-0.14671_21.20181009041228.bin)

```
Oct  9 09:12:15.363 UTC: %CLIENT_ORCH_LOG-6-CLIENT_ADDED_TO_RUN_STATE: Chassis 2 R0/0:
  wncd: Username entry (bob) joined with ssid (foo) for device with MAC: 1234.1234.5678
```

GUI Troubleshooting Dashboard



Troubleshooting

Logs

Manage Syslog, Webserver Log, License Log

Core Dump and System Report

View the list of core files and System Reports captured in the device

Debug Bundle

Capture require info like CLI outputs, logs as a single bundle for error reporting and debugging

Packet Capture

Capture packets with different filter options to feed into Wireshark for debugging

Ping and Trace Route

Check Ping-ability and Trace route info of a target destination through different sources

AP Packet Capture

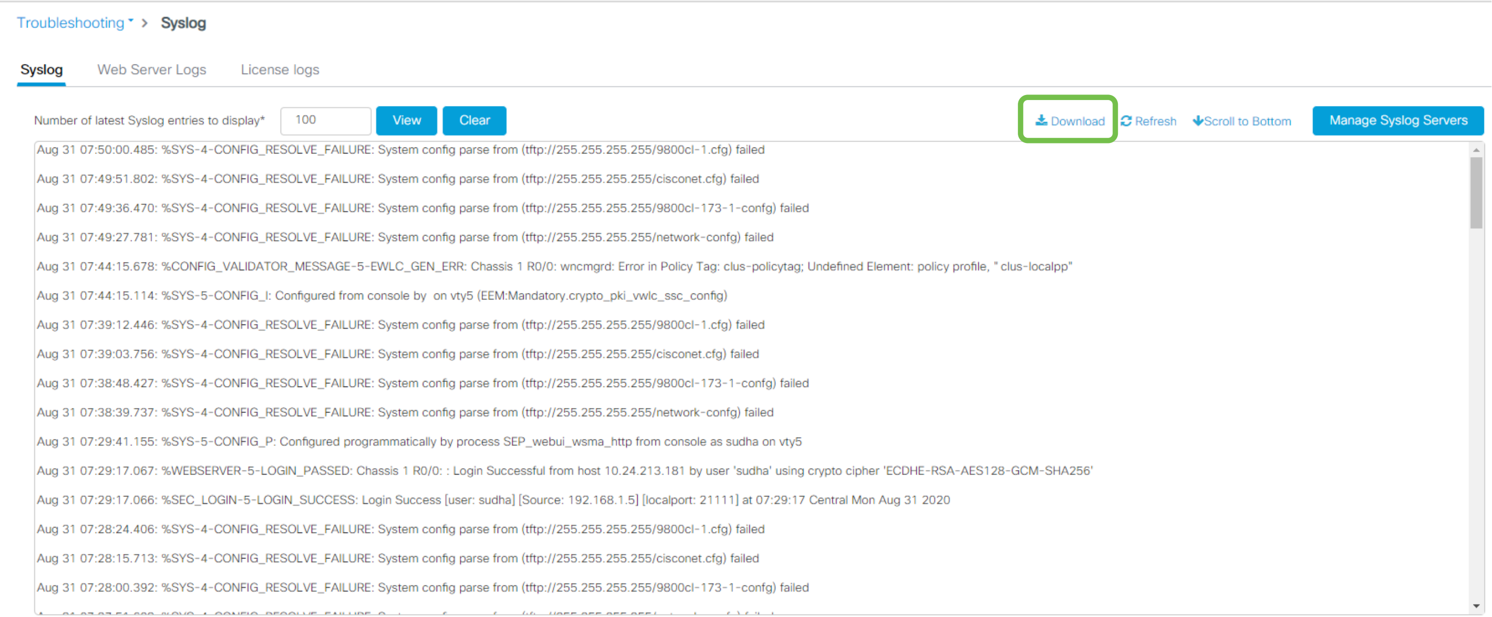
AP Packet Capture for troubleshooting wireless clients

Radioactive Trace

Collect conditional trace logs using MAC address of a Client, AP etc.

Syslogs - GUI

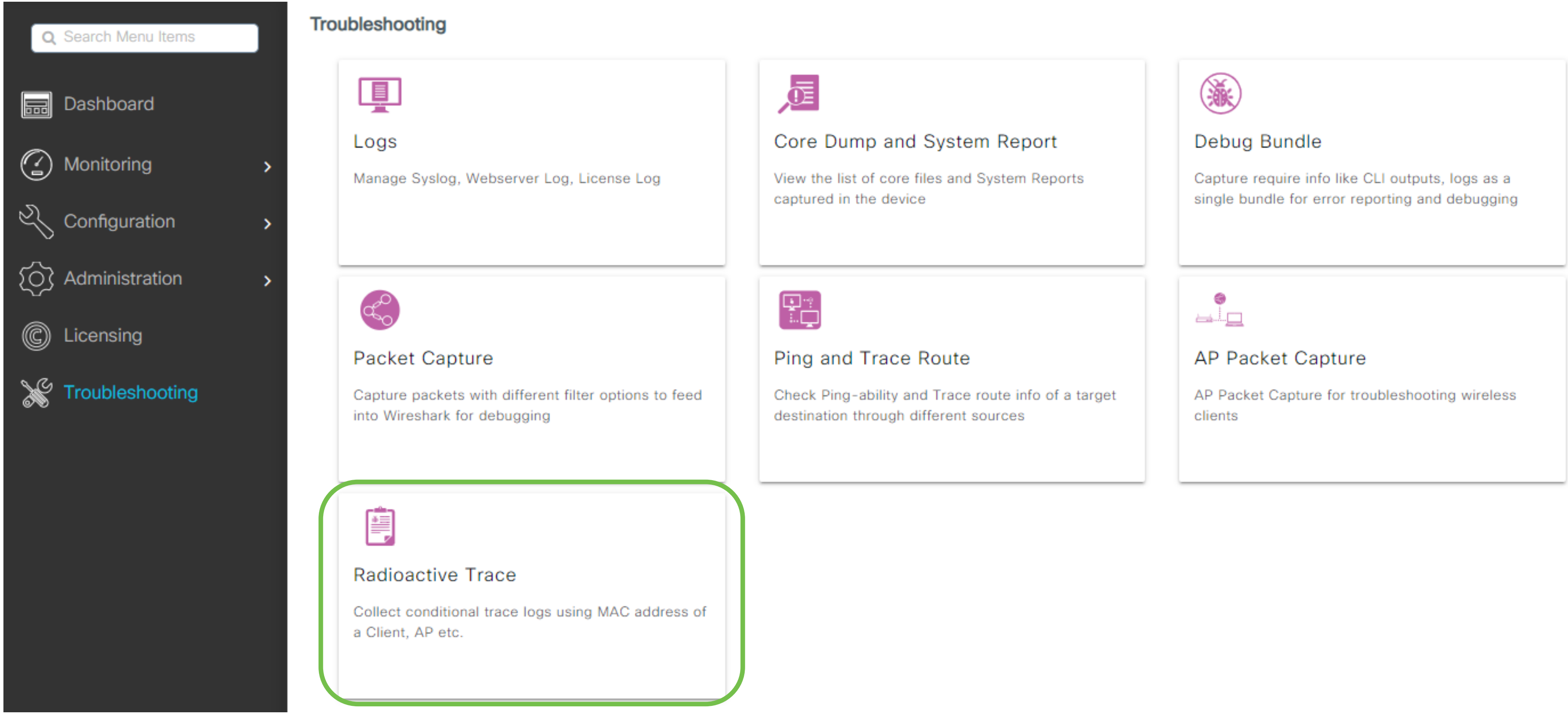
- Syslogs provide a quick view into any errors (trace level ERR) being reported by the system



```
Aug 31 07:50:00.485: %SYS-4-CONFIG_RESOLVE_FAILURE: System config parse from (tftp://255.255.255.255/9800cl-1.cfg) failed
Aug 31 07:49:51.802: %SYS-4-CONFIG_RESOLVE_FAILURE: System config parse from (tftp://255.255.255.255/cisconet.cfg) failed
Aug 31 07:49:36.470: %SYS-4-CONFIG_RESOLVE_FAILURE: System config parse from (tftp://255.255.255.255/9800cl-173-1-confg) failed
Aug 31 07:49:27.781: %SYS-4-CONFIG_RESOLVE_FAILURE: System config parse from (tftp://255.255.255.255/network-confg) failed
Aug 31 07:44:15.678: %CONFIG_VALIDATOR_MESSAGE-5-EWLC_GEN_ERR: Chassis 1 R0/0: wncmgrd: Error in Policy Tag: clus-policytag; Undefined Element: policy profile, " clus-localpp"
Aug 31 07:44:15.114: %SYS-5-CONFIG_I: Configured from console by  on vty5 (EEM:Mandatory.crypto_pki_vwlc_ssc_config)
Aug 31 07:39:12.446: %SYS-4-CONFIG_RESOLVE_FAILURE: System config parse from (tftp://255.255.255.255/9800cl-1.cfg) failed
Aug 31 07:39:03.756: %SYS-4-CONFIG_RESOLVE_FAILURE: System config parse from (tftp://255.255.255.255/cisconet.cfg) failed
Aug 31 07:38:48.427: %SYS-4-CONFIG_RESOLVE_FAILURE: System config parse from (tftp://255.255.255.255/9800cl-173-1-confg) failed
Aug 31 07:38:39.737: %SYS-4-CONFIG_RESOLVE_FAILURE: System config parse from (tftp://255.255.255.255/network-confg) failed
Aug 31 07:29:41.155: %SYS-5-CONFIG_P: Configured programmatically by process SEP_webui_wsma_http from console as sudha on vty5
Aug 31 07:29:17.067: %WEBSERVER-5-LOGIN_PASSED: Chassis 1 R0/0: : Login Successful from host 10.24.213.181 by user 'sudha' using crypto cipher 'ECDHE-RSA-AES128-GCM-SHA256'
Aug 31 07:29:17.066: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: sudha] [Source: 192.168.1.5] [localport: 21111] at 07:29:17 Central Mon Aug 31 2020
Aug 31 07:28:24.406: %SYS-4-CONFIG_RESOLVE_FAILURE: System config parse from (tftp://255.255.255.255/9800cl-1.cfg) failed
Aug 31 07:28:15.713: %SYS-4-CONFIG_RESOLVE_FAILURE: System config parse from (tftp://255.255.255.255/cisconet.cfg) failed
Aug 31 07:28:00.392: %SYS-4-CONFIG_RESOLVE_FAILURE: System config parse from (tftp://255.255.255.255/9800cl-173-1-confg) failed
```

Always-On Tracing (Default-On Tracing)

GUI Troubleshooting Dashboard



Troubleshooting

Logs

Manage Syslog, Webserver Log, License Log

Core Dump and System Report

View the list of core files and System Reports captured in the device

Debug Bundle

Capture require info like CLI outputs, logs as a single bundle for error reporting and debugging

Packet Capture

Capture packets with different filter options to feed into Wireshark for debugging

Ping and Trace Route

Check Ping-ability and Trace route info of a target destination through different sources

AP Packet Capture

AP Packet Capture for troubleshooting wireless clients

Radioactive Trace

Collect conditional trace logs using MAC address of a Client, AP etc.

Introducing Always On tracing

Contextual Logs WITHOUT enabling debugs

- Each process writes relevant events at Notice level
- No debug required
- Problem isolation assistance
  - Is client facing authentication issues or DHCP issue or something else
- Helps establish trends
  - Isolate if reported client connectivity problem is specific to certain APs or certain client mac addresses
- Box can store 48h approx. at max HW capacity, weeks typically

Always On tracing CLI

- Pre Process (in memory):

```
# show logging process <process daemon>
```

- Export to file:

```
# show logging process <process daemon> to-file <alwayson-processname.txt>
```

- Display in console:

```
# more bootflash:alwayson-processname.txt
```

- Export:

```
# copy bootflash:alwayson-processname.txt tftp://<serverip>/path OR
ftp://user:pass@serverip/path
```

Always On tracing – How to view

- Aggregated view across processes:

```
# show logging profile wireless filter {mac | ip} {client-mac | mobility-peer-ip}
to-file <alwayson-clientmac>.txt
```

- Focus on time window, export to file

```
# show logging profile wireless start timestamp “MM/DD/YYYY HH:MM:SS” filter mac
<mac addr> to-file <filename>
```

Default time in 16.12: since last boot

Default time starting 17.1 : last 10 minutes

- Focus last 5 minutes:

```
# show logging profile wireless start last 5 minutes
```

Always On: successful client connection

```
# show log profile wireless filter mac 0040.96b9.b5c4 to-file output.txt
```

```
[client-orch-sm] [21109]: (note): MAC: f0c1.f10b.8ac1  Association received. BSSID 7069.5a51.4ec0, old BSSID 0000.0000.0000, WLAN RomanTest, Slot
0 AP 7069.5a51.4ec0, AP4C77.6D9E.6162
[client-orch-state] [21109]: (note): MAC: f0c1.f10b.8ac1  Client state transition: S_CO_INIT -> S_CO_ASSOCIATING
[dot11] [21109]: (note): MAC: f0c1.f10b.8ac1  Association success. AID 1, Roaming = False, WGB = False, 11r = False, 11w = False AID list: 0x1|
0x0| 0x0| 0x0
[client-orch-state] [21109]: (note): MAC: f0c1.f10b.8ac1  Client state transition: S_CO_ASSOCIATING -> S_CO_L2_AUTH_IN_PROGRESS
[client-auth] [21109]: (note): MAC: f0c1.f10b.8ac1  ADD MOBILE sent. Client state flags: 0x71  BSSID: MAC: 7069.5a51.4ec0  capwap IFID:
0x90000004
[client-auth] [21109]: (note): MAC: f0c1.f10b.8ac1  L2 Authentication initiated. method DOT1X, Policy VLAN 1477,AAA override = 0 , NAC = 0
[ewlc-infra-evq] [21109]: (note): Authentication Success. Resolved Policy bitmap:11 for client f0c1.f10b.8ac1
[client-auth] [21109]: (note): MAC: f0c1.f10b.8ac1  L2 Authentication Key Exchange Start. Resolved VLAN: 1477, Audit Session id:
1E27300A0000000E127592C3
[client-keymgmt] [21109]: (note): MAC: f0c1.f10b.8ac1  EAP Key management successful. AKM:DOT1X Cipher:CCMP WPA2
[client-orch-sm] [21109]: (note): MAC: f0c1.f10b.8ac1  Mobility discovery triggered. Client mode: Local
[client-orch-state] [21109]: (note): MAC: f0c1.f10b.8ac1  Client state transition: S_CO_L2_AUTH_IN_PROGRESS ->
S_CO_MOBILITY_DISCOVERY_IN_PROGRESS
[mm-client] [21109]: (note): MAC: f0c1.f10b.8ac1  Mobility Successful. Roam Type None, Sub Roam Type MM_SUB_ROAM_TYPE_NONE, Previous BSSID MAC:
0000.0000.0000   Client IFID: 0xa0000001, Client Role: Local PoA: 0x90000004 PoP: 0x0
[client-auth] [21109]: (note): MAC: f0c1.f10b.8ac1  ADD MOBILE sent. Client state flags: 0x72  BSSID: MAC: 7069.5a51.4ec0  capwap IFID:
0x90000004
[client-orch-state] [21109]: (note): MAC: f0c1.f10b.8ac1  Client state transition: S_CO_MOBILITY_DISCOVERY_IN_PROGRESS ->
S_CO_DPATH_PLUMB_IN_PROGRESS
[dot11] [21109]: (note): MAC: f0c1.f10b.8ac1  Client datapath entry params - ssid:RomanTest,slot_id:0 bssid ifid: 0x0, radio_ifid: 0x90000003,
wlan_ifid: 0xf0400002
[dpath_svc] [21109]: (note): MAC: f0c1.f10b.8ac1  Client datapath entry created for ifid 0xa0000001
[client-orch-state] [21109]: (note): MAC: f0c1.f10b.8ac1  Client state transition: S_CO_DPATH_PLUMB_IN_PROGRESS -> S_CO_IP_LEARN_IN_PROGRESS
[client-iplearn] [21109]: (note): MAC: f0c1.f10b.8ac1  Client IP learn successful. Method: DHCP IP: 192.168.77.200
[client-orch-state] [21109]: (note): MAC: f0c1.f10b.8ac1  Client state transition: S_CO_IP_LEARN_IN_PROGRESS -> S_CO_RUN
```

Always On: client connection failure

```
#sh logging profile wireless filter mac f0c1.f10b.8ac to-file dot1x-failure.txt
```

```
2019/10/29 09:35:34.048 {wncd_x_R0-0}{1}: [client-orch-sm] [19470]: (note): MAC: f0c1.f10b.8ac1  Association received. BSSID 7069.5a51.4ec0, old BSSID
7069.5a51.4ec0, WLAN RomanTest, Slot 0 AP 7069.5a51.4ec0, AP4C77.6D9E.6162

2019/10/29 09:35:34.048 {wncd_x_R0-0}{1}: [client-orch-state] [19470]: (note): MAC: f0c1.f10b.8ac1  Client state transition: S_CO_L2_AUTH_IN_PROGRESS
-> S_CO_L2_AUTH_IN_PROGRESS

2019/10/29 09:35:34.048 {wncd_x_R0-0}{1}: [dot11] [19470]: (note): MAC: f0c1.f10b.8ac1  Association success. AID 1, Roaming = False, WGB = False, 11r
= False, 11w = False AID list: 0x1| 0x0| 0x0| 0x0

2019/10/29 09:35:34.048 {wncd_x_R0-0}{1}: [client-orch-state] [19470]: (note): MAC: f0c1.f10b.8ac1  Client state transition: S_CO_L2_AUTH_IN_PROGRESS
-> S_CO_L2_AUTH_IN_PROGRESS

2019/10/29 09:35:34.048 {wncd_x_R0-0}{1}: [client-auth] [19470]: (note): MAC: f0c1.f10b.8ac1  ADD MOBILE sent. Client state flags: 0x71  BSSID: MAC:
7069.5a51.4ec0  capwap IFID: 0x90000004

2019/10/29 09:35:34.051 {wncd_x_R0-0}{1}: [client-auth] [19470]: (note): MAC: f0c1.f10b.8ac1  L2 Authentication initiated. method DOT1X, Policy VLAN
1477,AAA override = 0 , NAC = 0

2019/10/29 09:35:34.330 {wncd_x_R0-0}{1}: [errmsg] [19470]: (note): %DOT1X-5-FAIL: Authentication failed for client (f0c1.f10b.8ac1) with reason (Cred
Fail) on Interface capwap_90000004 AuditSessionID 000000000000000B16D9A13D Username: drghgdf
```

Always On : AP join failures

```
# show log profile wir filter mac <ap radio mac> to-file output.txt
```

- Unsupported AP

```
[apmgr-capwap-join] [1263]: UUID: 0, ra: 0, TID: 0 (ERR): d824.bde8.3690 Join request not accepted: Unsupported AP
Model AIR-LAP1142N-A-K9
```

- Reg Domain failure

```
[apmgr-capwap-config] [1394]: UUID: 10000000002ed, (ERR): f44e.0597.fb50 Failed to verify reg domain slot.
validation of country code(UX) to regulatory domain(-A) error:1
[apmgr-capwap-config] [1394]: UUID: 10000000002ed, (ERR): f44e.0597.fb50 Failed to get ap default country code. Get
default country code for AP error.
[apmgr-capwap-config] [1394]: UUID: 10000000002ed, (ERR): f44e.0597.fb50 Failed to set reg domain check status.
country code US  is not configured on WLC
```

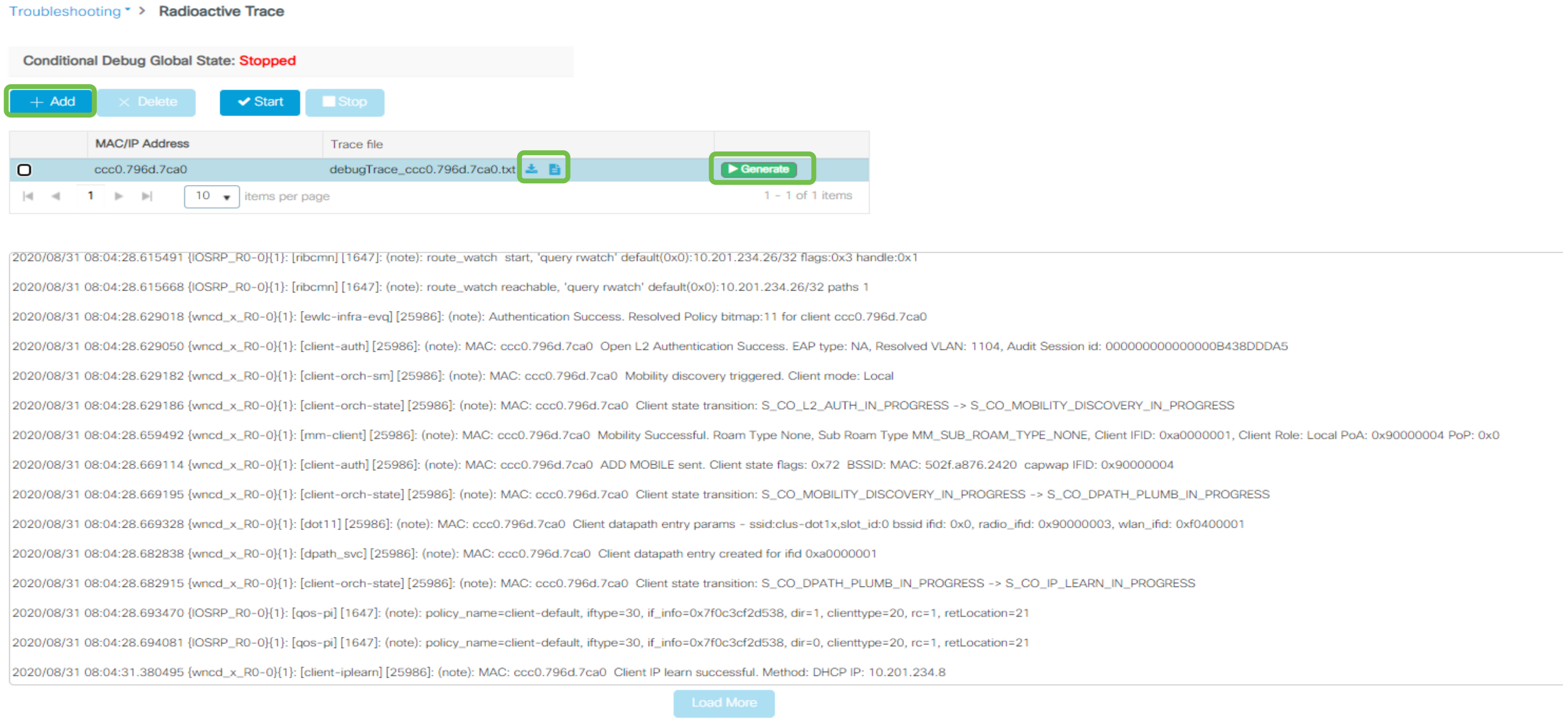
- Cert Failure

```
[apmgr-capwap-config] [1394]: UUID: 10000000002ed, (ERR), %PKI-3-CERTIFICATE_INVALID_NOT_YET_VALID: Certificate
chain validation has failed.  The certificate (SN: 6B4F09560000001763DF) is not yet valid   Validity period starts
on 22:48:43 IST Sep 9 2014
```

- Discovery to non wireless mgmt interface

```
{wncmgrd_R0-0}{2}: [capwapac-srvr] [16320]: UUID: 0, ra::0, TID: 0 (ERR): IP:3.3.3.1[5246], Discovery to non
wireless mgmt interface
```

Always on Tracing - GUI



Conditional Debug Global State: Stopped

	MAC/IP Address	Trace file	
□	ccc0.796d.7ca0	debugTrace_ccc0.796d.7ca0.txt	▶ Generate

```
2020/08/31 08:04:28.615491 {IOSRP_R0-0}{1}: [ribcmn] [1647]: (note): route_watch  start, 'query rwatch' default(0x0):10.201.234.26/32 flags:0x3 handle:0x1
2020/08/31 08:04:28.615668 {IOSRP_R0-0}{1}: [ribcmn] [1647]: (note): route_watch reachable, 'query rwatch' default(0x0):10.201.234.26/32 paths 1
2020/08/31 08:04:28.629018 {wncd_x_R0-0}{1}: [ewlc-infra-evq] [25986]: (note): Authentication Success. Resolved Policy bitmap:11 for client ccc0.796d.7ca0
2020/08/31 08:04:28.629050 {wncd_x_R0-0}{1}: [client-auth] [25986]: (note): MAC: ccc0.796d.7ca0  Open L2 Authentication Success. EAP type: NA, Resolved VLAN: 1104, Audit Session id: 000000000000000B438DDDA5
2020/08/31 08:04:28.629182 {wncd_x_R0-0}{1}: [client-orch-sm] [25986]: (note): MAC: ccc0.796d.7ca0  Mobility discovery triggered. Client mode: Local
2020/08/31 08:04:28.629186 {wncd_x_R0-0}{1}: [client-orch-state] [25986]: (note): MAC: ccc0.796d.7ca0  Client state transition: S_CO_L2_AUTH_IN_PROGRESS -> S_CO_MOBILITY_DISCOVERY_IN_PROGRESS
2020/08/31 08:04:28.659492 {wncd_x_R0-0}{1}: [mm-client] [25986]: (note): MAC: ccc0.796d.7ca0  Mobility Successful. Roam Type None, Sub Roam Type MM_SUB_ROAM_TYPE_NONE, Client IFID: 0xa0000001, Client Role: Local PoA: 0x90000004 PoP: 0x0
2020/08/31 08:04:28.669114 {wncd_x_R0-0}{1}: [client-auth] [25986]: (note): MAC: ccc0.796d.7ca0  ADD MOBILE sent. Client state flags: 0x72  BSSID: MAC: 502f.a876.2420  capwap IFID: 0x90000004
2020/08/31 08:04:28.669195 {wncd_x_R0-0}{1}: [client-orch-state] [25986]: (note): MAC: ccc0.796d.7ca0  Client state transition: S_CO_MOBILITY_DISCOVERY_IN_PROGRESS -> S_CO_DPATH_PLUMB_IN_PROGRESS
2020/08/31 08:04:28.669328 {wncd_x_R0-0}{1}: [dot11] [25986]: (note): MAC: ccc0.796d.7ca0  Client datapath entry params - ssid:clus-dot1x,slot_id:0 bssid ifid: 0x0, radio_ifid: 0x90000003, wlan_ifid: 0xf0400001
2020/08/31 08:04:28.682838 {wncd_x_R0-0}{1}: [dpath_svc] [25986]: (note): MAC: ccc0.796d.7ca0  Client datapath entry created for ifid 0xa0000001
2020/08/31 08:04:28.682915 {wncd_x_R0-0}{1}: [client-orch-state] [25986]: (note): MAC: ccc0.796d.7ca0  Client state transition: S_CO_DPATH_PLUMB_IN_PROGRESS -> S_CO_IP_LEARN_IN_PROGRESS
2020/08/31 08:04:28.693470 {IOSRP_R0-0}{1}: [qos-pi] [1647]: (note): policy_name=client-default, iftype=30, if_info=0x7f0c3cf2d538, dir=1, clienttype=20, rc=1, retLocation=21
2020/08/31 08:04:28.694081 {IOSRP_R0-0}{1}: [qos-pi] [1647]: (note): policy_name=client-default, iftype=30, if_info=0x7f0c3cf2d538, dir=0, clienttype=20, rc=1, retLocation=21
2020/08/31 08:04:31.380495 {wncd_x_R0-0}{1}: [client-iplearn] [25986]: (note): MAC: ccc0.796d.7ca0  Client IP learn successful. Method: DHCP IP: 10.201.234.8
```

Trace-on-Failure

Trace-on-Failure (TOF)
(Not fully supported until 17.3)

- 55 Predefined failure codes tracked
- Available stats

```
# show wireless stats trace-on-failure
001. AP radio reset....................................: 0
002. AP reset..........................................: 0
003. Client disjoin due to AP radio reset...............: 0
004. Client disjoin due to AP reset.....................: 0
005. Export client MMIF................................: 0
006. Export client MM..................................: 0
007. Export client generic.............................: 0
011. AP join failure...................................: 0
012. AP initial configuration failure..................: 
44335
```

Trace on Failure Summary

- You can see indexed recent failures. This not only gives you a quick failure of recent failure but includes timestamp and UUID which can then be used to look at the section of trace logs to get additional context of failure.

```
[16.12]# show logging trace-on-failure summary
[17.1]# show logging profile wireless (filter mac) trace-on-failure
```

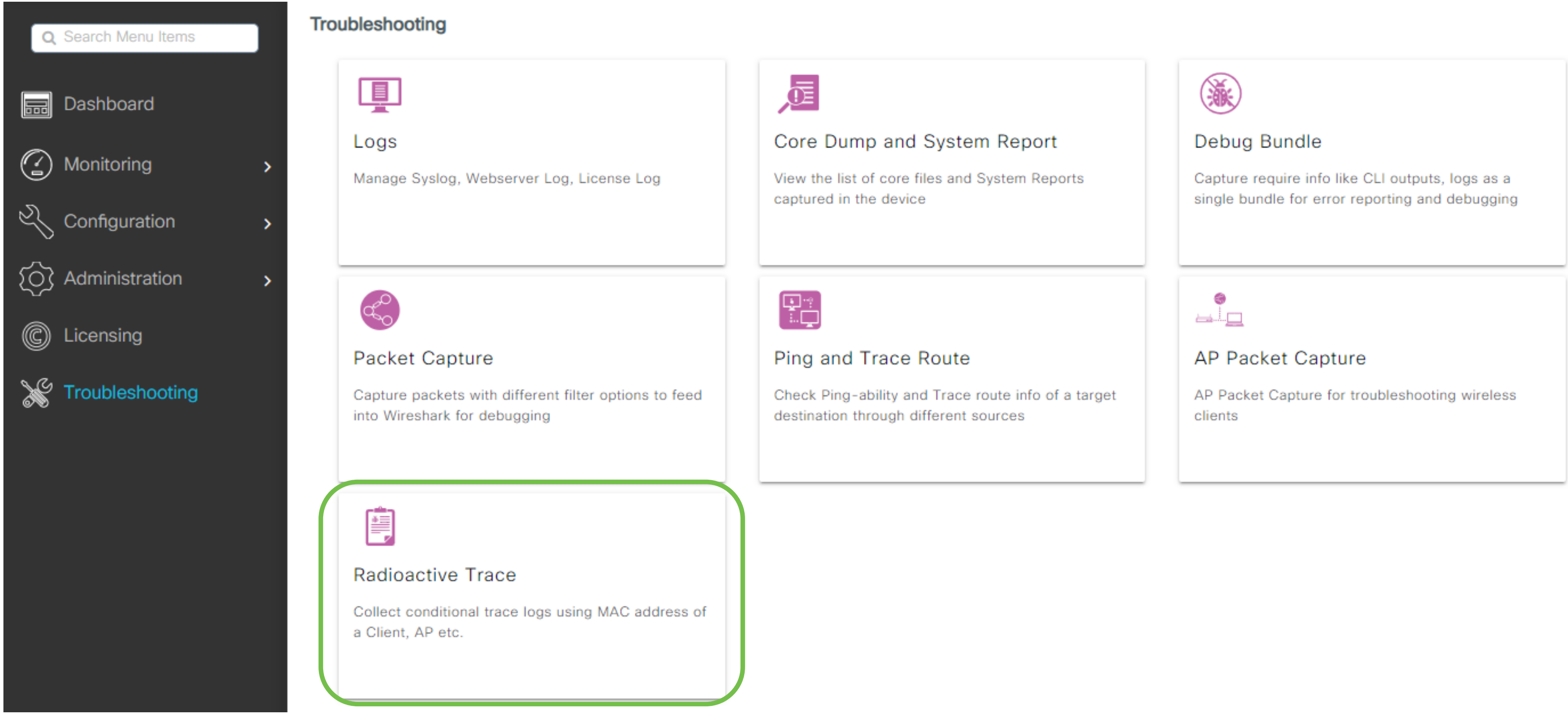
```
Time                                   UUID                          Log
---------------------------------------------------------------------------------------------
2018/09/21 04:43:52.773    0x1000000004c93    2048.2000.0300 AP_CFG_STATUS_FAIL : Apmgr failure reason : Regulatory
2018/09/21 04:43:52.990    0x1000000004cbf     2048.2000.0500 AP_CFG_STATUS_FAIL : Apmgr failure reason : Regulatory
2018/09/21 04:43:52.999    0x1000000004ced e836.171f.a162 CLIENT_STAGE_TIMEOUT State = IP_LEARNING, WLAN profile = 
ACLtest, Policy profile = leap, AP name = LABap_2802
2018/09/21 04:43:53.068    0x1000000004ce5    2048.2000.0200 AP_CFG_STATUS_FAIL : Apmgr failure reason : Regulatory
2018/09/21 04:43:53.226    0x1000000004d05    2048.2000.0700 AP_CFG_STATUS_FAIL : Apmgr failure reason : Regulatory
2018/09/21 04:43:53.270    0x1000000004d17    2048.2000.0600 AP_CFG_STATUS_FAIL : Apmgr failure reason : Regulatory
2018/09/21 04:43:55.626    0x1000000004e61    2048.2000.1200 AP_CFG_STATUS_FAIL : Apmgr failure reason : Regulatory
2018/12/12 12:26:35.406    0x10000000cd09b    8875.56c6.f000 AP_JOIN_FAIL : Apmgr failure reason : Unsupported ap,
2018/12/17 13:18:32.097    0x10000002c7428    08cc.68b4.4660 CAPWAPAC_HEARTBEAT_EXPIRY
```

Trace on Failure Details

```
# show log profile wir filter uuid 0x10000000cd09b to-file <filename>

# more bootflash:<filename>

2018/12/12 12:26:35.406 {wncd_x_R0-3}{1}: [ewlc-infra-evq] [3862]: (note): Data type :  Message handle
2018/12/12 12:26:35.406 {wncd_x_R0-3}{1}: [apmgr-capwap-join] [3862]: (ERR): 8875.56c6.f000 Join request
not accepted: Unsupported AP Model AIR-CAP3602I-E-K9
2018/12/12 12:26:35.406 {wncd_x_R0-3}{1}: [apmgr-capwap-join] [3862]: (ERR): 8875.56c6.f000 Failed to
process join request. Unable to decode apmgr join response
2018/12/12 12:26:35.406 {wncd_x_R0-3}{1}: [apmgr-ap-global] [3862]: (ERR): 8875.56c6.f000 Failed to handle
ap sm join request. Unable to process apmgr join request
2018/12/12 12:26:35.406 {wncd_x_R0-3}{1}: [ewlc-infra-evq] [3862]: (ERR): 8875.56c6.f000 AP_JOIN_FAIL :
Apmgr failure reason : Unsupported ap, Policy tag : , Site tag : , Rf tag : default-rf-tag
2018/12/12 12:26:35.406 {wncd_x_R0-3}{1}: [apmgr-db] [3862]: (ERR): Failed to get ap name mac map record
for delete. Name: AP3602I-E-K9. Reason: No such file or directory
2018/12/12 12:26:35.406 {wncd_x_R0-3}{1}: [apmgr-db] [3862]: (ERR): 8875.56c6.f000 Delete ap name map
record from the apmgr failed: 2
2018/12/12 12:26:35.406 {wncd_x_R0-3}{1}: [capwapac-smgr-sess-fsm] [3862]: (ERR): Session-IP:
192.168.17.146[57187] Mac: 8875.56c6.f000 Unmapped previous state in transition S_JOIN_PROCESS to S_END on
E_AP_INTERFACE_DOWN
```

Radioactive tracing (Conditional Debuggin

GUI Troubleshooting Dashboard



Troubleshooting

Logs

Manage Syslog, Webserver Log, License Log

Core Dump and System Report

View the list of core files and System Reports captured in the device

Debug Bundle

Capture require info like CLI outputs, logs as a single bundle for error reporting and debugging

Packet Capture

Capture packets with different filter options to feed into Wireshark for debugging

Ping and Trace Route

Check Ping-ability and Trace route info of a target destination through different sources

AP Packet Capture

AP Packet Capture for troubleshooting wireless clients

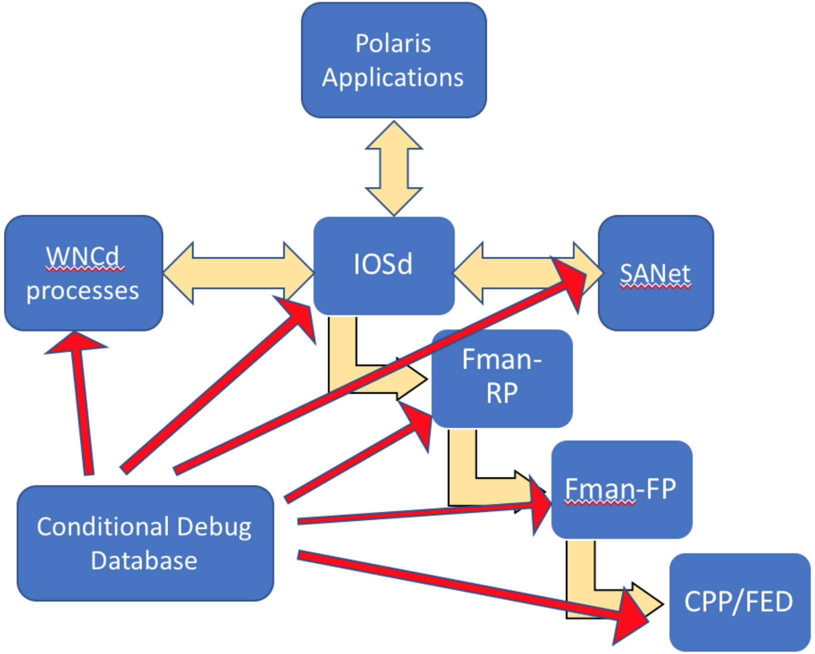
Radioactive Trace

Collect conditional trace logs using MAC address of a Client, AP etc.

Radioactive tracing

Building on existing conditional debugging CLI

- Collect additional data about a particular IP or mac
- Roughly similar to the old “deb client mac”
- Formally: it is “store and display” process
- Filter needs to match available info
- A lot more detailed
  - Always On: 18 lines
  - Radioactive: 180 lines
  - Radioactive + Internal: 1800 lines



Radioactive tracing

Difficult!

```
# debug platform condition feature wireless mac <client mac>
# debug platform condition start
```

(reproduce issue)

```
# debug platform condition stop
```

```
# show logging profile wireless  [(start timestamp “Date&time”) level debug filter mac
<client mac> to-file <filename>]
```

```
# more flash:<filename>
```

Radioactive Tracing: Easy way…

Use this

- Macro to collect and export in one go:

```
#  debug wireless mac <mac-of-client> ftp-server ser.ver.ip.add /directory
```

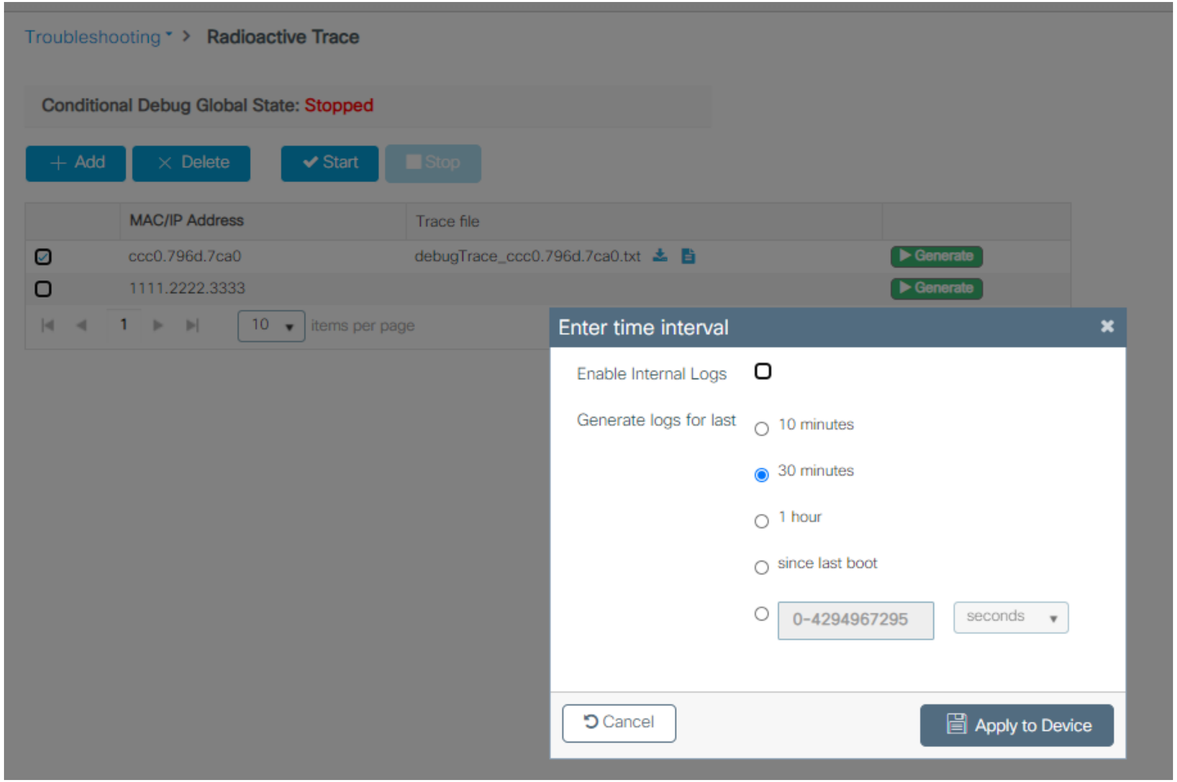
- Runs for 30 min, or set a timer
- Stop with

```
# no debug wireless mac <mac-of-client>
```

- Destination can be FTP or File (flash)
  - File is more reliable
  - FTP needs write access, previous config

Radioactive Tracing: Even Easier...

Use this!



Troubleshooting > Radioactive Trace

Conditional Debug Global State: Stopped

	MAC/IP Address	Trace file	
☑	ccc0.796d.7ca0	debugTrace_ccc0.796d.7ca0.txt	▶ Generate
□	1111.2222.3333		▶ Generate

Enter time interval

Enable Internal Logs □

Generate logs for last
○ 10 minutes
● 30 minutes
○ 1 hour
○ since last boot
○ 0-4294967295 seconds

Radioactive Tracing Filtering and Cleanup

- AP debugging by mac works for all radio/rrm/etc processes. DTLS will not work
- AP debugging by its IP address works for DTLS, but misses all later processes

Remember: set filter for the desired context

- **Always remove conditions**

  # clear platform condition all

  # undebug all

Process Tracing (Unconditional Debugging)

Process Daemon Specific Debugging

Unconditional Debugging

- Single process focused troubleshooting
  - Examples: RRM, nginx web server

- To view current log level set for a process trace

```
# show platform software trace level <rrm-mgrd | wncd | mobility>
chassis active R0
```

Process Daemon Specific Debugging

- Enable:

```
# set platform software trace <rrm-mgrd | nginx | nmspd> chassis active R0 all debug
```

(reproduce issue)

- Collect traces:

```
# show logging process <rrm-mgrd | nginx | nmspd> to-file <debugtrace-rrmd.txt>
```

- View:

```
# more bootflash:debugtrace-rrmd.txt
```

- Export:

```
# copy bootflash:debugtrace-rrmd.txt { tftp:, ftp:, http:, https:, scp: }
```

- Disable:

```
# undebug all OR # set platform software trace <> chassis active R0 all notice
```

Process Daemon Specific Debugging for mDNS

- Enable:

```
# set platform software trace wncd 0 chassis active r0 mdns verbose
# sh platform software trace level wncd 0 chassis active R0 | in Verbose
```

(reproduce issue)

- Collect traces:

```
#  sh platform software trace message wncd 0 chassis active R0
```

- Disable:

```
# undebug all OR # set platform software trace <> chassis active R0 all notice
```

Process Daemon Specific Debugging for mDNS

- **Successful service learning:**

```
2019/11/08 07:37:06.976 {wncd_x_R0-0}{1}: [mdns] [28837]: (verbose): Received READ Callback for IPV4 mDNS
packet
…
2019/11/08 07:37:06.975 {wncd_x_R0-0}{1}: [mdns] [28837]: (debug): MDNS_ADVT:[MAC:88e9.fe7a.04c8]TXT record
added/updated sucessfully : Nxxxxx-M-X2HX._airplay._tcp.local
…
2019/11/08 07:37:06.975 {wncd_x_R0-0}{1}: [mdns] [28837]: (debug): MDNS_ADVT:[MAC:88e9.fe7a.04c8]TXT record
added/updated sucessfully : Nxxxxx-M-X2HX._airserver._tcp.local
```

- **Failed mDNS processing:**

```
2019/11/08 07:37:36:50.711 {wncd_x_R0-0}{1}: [mdns] [26786]: (debug): Received READ Callback for IPV4 mDNS
packet

2019/11/08 07:37:36:50.711 {wncd_x_R0-0}{1}: [mdns] [26786]: (verbose): In ret_buffer pak: 0x55bd04ee9ff8
bpak->buffer_start 0x55bd04ee1098 bpak->subblock 0x0

2019/11/08 07:37:36:50.711 {wncd_x_R0-0}{1}: [mdns] [26786]: (verbose): MDNS record Search: record with
wlan_id: 2 found

2019/11/08 07:37:36:50.711 {wncd_x_R0-0}{1}: [mdns] [26786]: (verbose): Dropping mDNS packet, SVI interface
(VLAN : 1477) not present/UP
```

Tracing Summary - What is what?

IOSd Logging

- Your Traditional Syslog

Binary Tracing

- Fast infrastructure for real-time logging

Always On Tracing

- Real time data collection for all relevant events

Conditional Debugging/Radioactive Tracing

- Per IP/MAC address debugging

Non-Conditional Debugging/Per Process Tracing

- Your traditional debug

Tracing Summary – When to use?

Basic client/AP data collection:

- Data is there, just pull it…
- Collect data with “`show logging profile wireless filter {mac | ip}`”..

Advanced client/AP:

- Use Radioactive Tracing
- Collect data with “`debug wireless mac <mac-of-client> ftp-server ser.ver.ip.add /directory`”

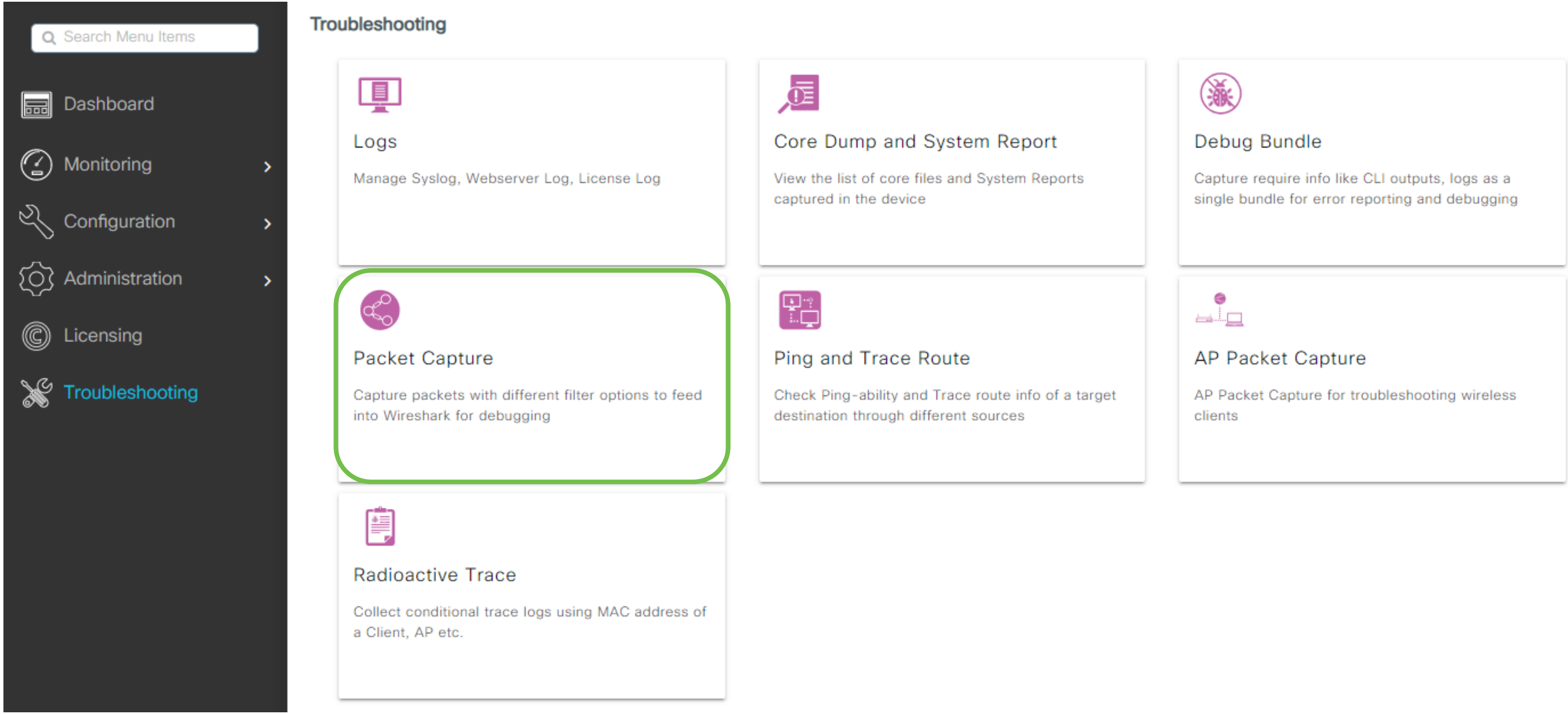
Basic Box logs

- Traditional show logs/syslog

Embedded Packet Capture

WIRESHARK®

GUI Troubleshooting Dashboard



Search Menu Items

Dashboard

Monitoring ›

Configuration ›

Administration ›

Licensing

Troubleshooting

Troubleshooting

Logs

Manage Syslog, Webserver Log, License Log

Core Dump and System Report

View the list of core files and System Reports captured in the device

Debug Bundle

Capture require info like CLI outputs, logs as a single bundle for error reporting and debugging

Packet Capture

Capture packets with different filter options to feed into Wireshark for debugging

Ping and Trace Route

Check Ping-ability and Trace route info of a target destination through different sources

AP Packet Capture

AP Packet Capture for troubleshooting wireless clients

Radioactive Trace

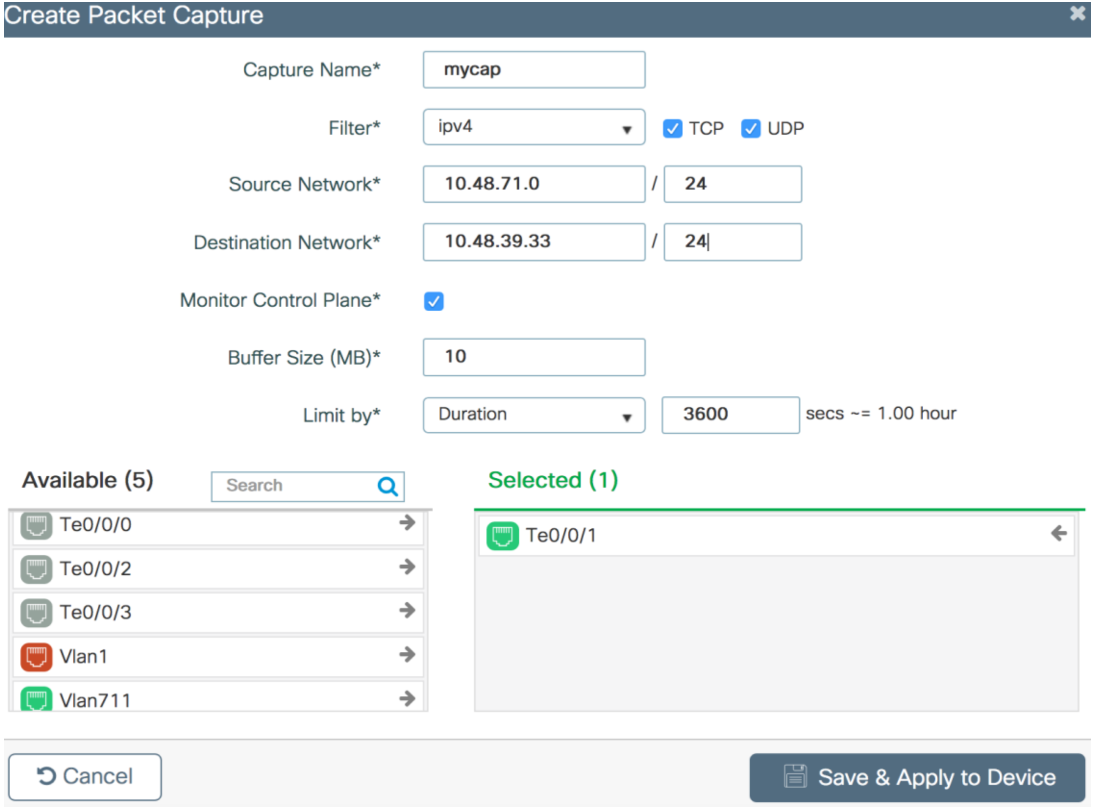
Collect conditional trace logs using MAC address of a Client, AP etc.

Embedded Packet Capture

- Get packets sent from or to and through the controller
- Export to Wireshark
- No need for switch capture
- Accessible either from GUI or CLI

Embedded Packet Capture (EPC) web interface

- Web interface to the existing EPC CLI “monitor capture …”
- One click start/stop/download
- Physical and VLAN interfaces can be selected



Create Packet Capture

Capture Name*: mycap

Filter*: ipv4 ☑ TCP ☑ UDP

Source Network*: 10.48.71.0 / 24

Destination Network*: 10.48.39.33 / 24

Monitor Control Plane*: ☑

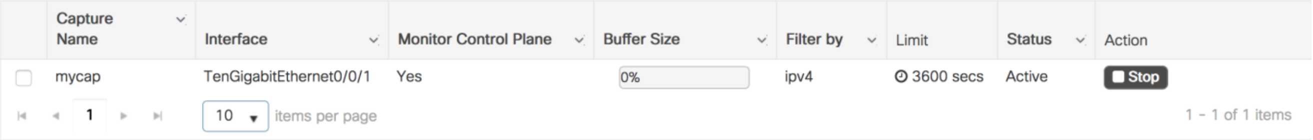
Buffer Size (MB)*: 10

Limit by*: Duration 3600 secs ~= 1.00 hour

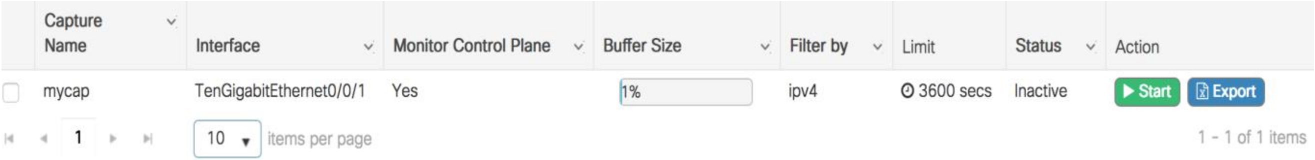
Available (5)	Selected (1)
Te0/0/0	Te0/0/1
Te0/0/2	
Te0/0/3	
Vlan1	
Vlan711	

Embedded Packet Capture web interface

One click start/stop/download



	Capture Name	Interface	Monitor Control Plane	Buffer Size	Filter by	Limit	Status	Action
□	mycap	TenGigabitEthernet0/0/1	Yes	0%	ipv4	3600 secs	Active	Stop



	Capture Name	Interface	Monitor Control Plane	Buffer Size	Filter by	Limit	Status	Action
□	mycap	TenGigabitEthernet0/0/1	Yes	1%	ipv4	3600 secs	Inactive	Start Export

Embedded Packet Capture CLI

```
monitor capture test interface GigabitEthernet2 both
monitor capture test control-plane both
monitor capture test match any
monitor capture test buffer size 100 circular
monitor capture test limit pps 1000000
monitor capture test start
monitor capture test stop
monitor capture test export bootflash:test.pcap
```

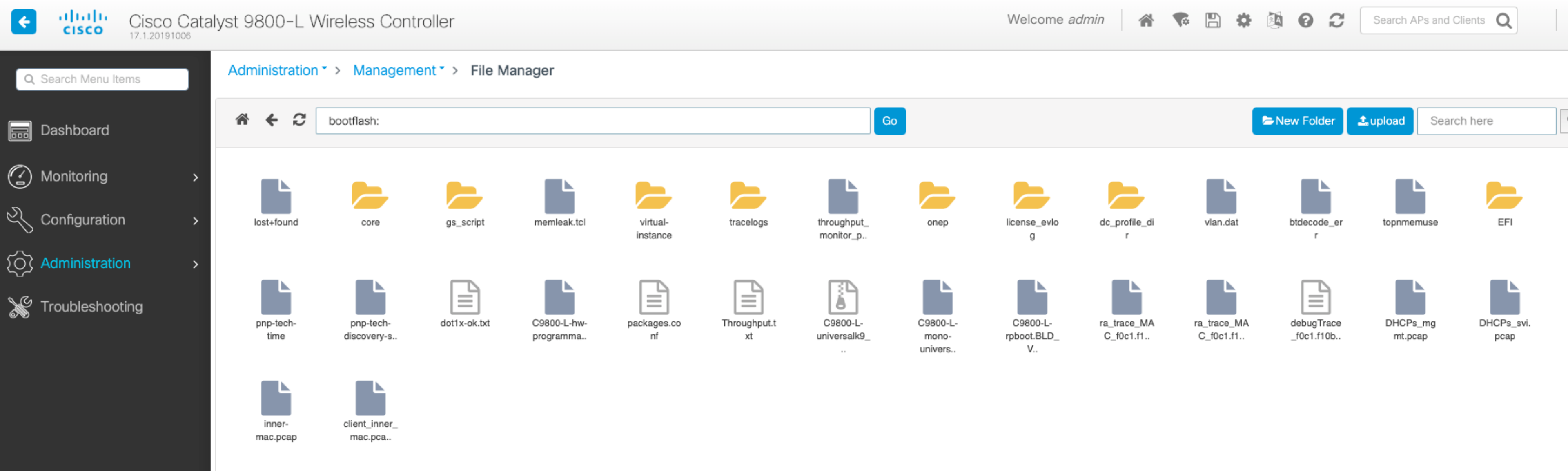
EPC CLI – Granular Filtering Options

- While GUI provides ease of use, it can only filter for an ipv4/ipv6 address.
- For more granular filtering using access-list etc, CLI is preferable.
- With 16.x, in order to capture traffic for one client, it has to be filtered on ip address of AP, it was registering to.
- With 17.x, we have an additional filter to match inner identity (currently *mac-address* only) which allows to focus on traffic related to specific client when CAPWAP encapsulated.

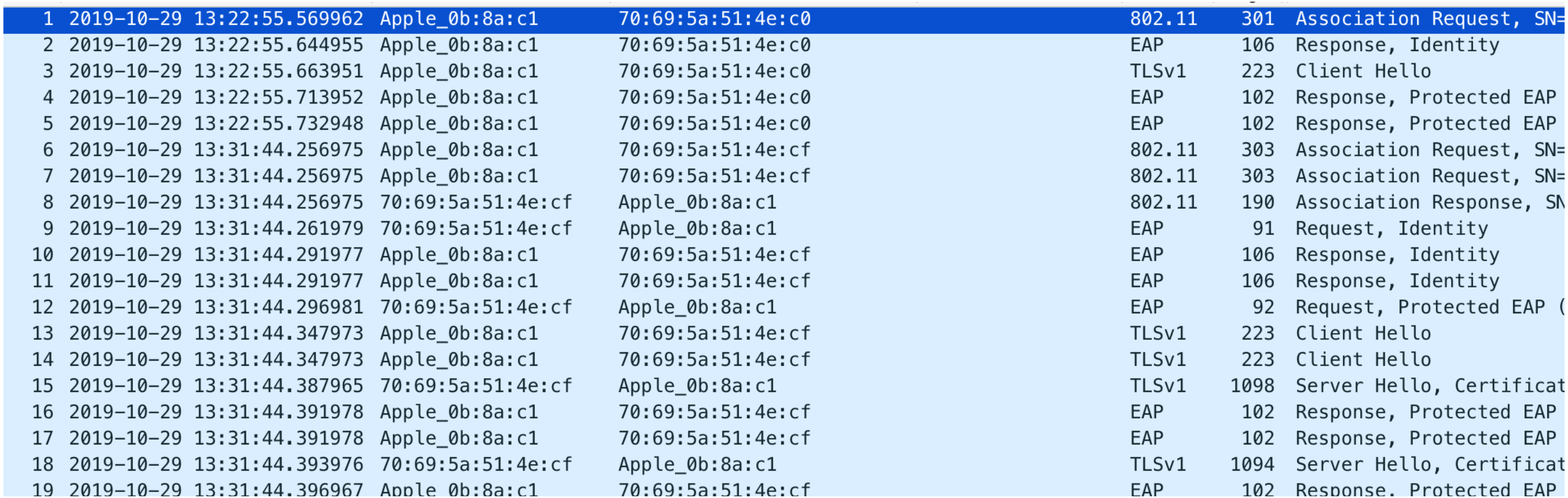
```
# monitor capture client_inner_mac inner mac f0c1.f10b.8ac1 interface vlan39 both control-plane
both
# monitor capture client_inner_mac match any
# monitor capture client_inner_mac start
# monitor capture client_inner_mac stop
# monitor capture client_inner_mac export bootflash:inner-mac.pcap
```

Embedded Packet Capture CLI

Collected captures can be either uploaded to some file server in the network or downloaded from WLC web interface directly.



EPC CLI – Granular Filtering Options



```
 1 2019-10-29 13:22:55.569962 Apple_0b:8a:c1       70:69:5a:51:4e:c0    802.11  301 Association Request, SN=
 2 2019-10-29 13:22:55.644955 Apple_0b:8a:c1       70:69:5a:51:4e:c0    EAP     106 Response, Identity
 3 2019-10-29 13:22:55.663951 Apple_0b:8a:c1       70:69:5a:51:4e:c0    TLSv1   223 Client Hello
 4 2019-10-29 13:22:55.713952 Apple_0b:8a:c1       70:69:5a:51:4e:c0    EAP     102 Response, Protected EAP
 5 2019-10-29 13:22:55.732948 Apple_0b:8a:c1       70:69:5a:51:4e:c0    EAP     102 Response, Protected EAP
 6 2019-10-29 13:31:44.256975 Apple_0b:8a:c1       70:69:5a:51:4e:cf    802.11  303 Association Request, SN=
 7 2019-10-29 13:31:44.256975 Apple_0b:8a:c1       70:69:5a:51:4e:cf    802.11  303 Association Request, SN=
 8 2019-10-29 13:31:44.256975 70:69:5a:51:4e:cf    Apple_0b:8a:c1       802.11  190 Association Response, SN
 9 2019-10-29 13:31:44.261979 70:69:5a:51:4e:cf    Apple_0b:8a:c1       EAP      91 Request, Identity
10 2019-10-29 13:31:44.291977 Apple_0b:8a:c1       70:69:5a:51:4e:cf    EAP     106 Response, Identity
11 2019-10-29 13:31:44.291977 Apple_0b:8a:c1       70:69:5a:51:4e:cf    EAP     106 Response, Identity
12 2019-10-29 13:31:44.296981 70:69:5a:51:4e:cf    Apple_0b:8a:c1       EAP      92 Request, Protected EAP (
13 2019-10-29 13:31:44.347973 Apple_0b:8a:c1       70:69:5a:51:4e:cf    TLSv1   223 Client Hello
14 2019-10-29 13:31:44.347973 Apple_0b:8a:c1       70:69:5a:51:4e:cf    TLSv1   223 Client Hello
15 2019-10-29 13:31:44.387965 70:69:5a:51:4e:cf    Apple_0b:8a:c1       TLSv1  1098 Server Hello, Certificat
16 2019-10-29 13:31:44.391978 Apple_0b:8a:c1       70:69:5a:51:4e:cf    EAP     102 Response, Protected EAP
17 2019-10-29 13:31:44.391978 Apple_0b:8a:c1       70:69:5a:51:4e:cf    EAP     102 Response, Protected EAP
18 2019-10-29 13:31:44.393976 70:69:5a:51:4e:cf    Apple_0b:8a:c1       TLSv1  1094 Server Hello, Certificat
19 2019-10-29 13:31:44.396967 Apple_0b:8a:c1       70:69:5a:51:4e:cf    EAP     102 Response, Protected EAP
```

Data Plane Packet Tracer

Data Plane Packet Tracing

- Data plane “view” of specified traffic
- Collect X packets, and dump information
- Verify which features are processing each frame

- It is not a packet capture -> EPC
- Mostly IP related traffic (no wireless info)

Packet Tracing

- Set condition

```
# debug platform condition mac 001e.e5e2.35cf both
```

Enable conditional debugging

```
# debug platform start
```

- Verify enabled conditions

```
# show platform conditions
```

- Enable packet-tracer and specify the number of packets to collect

```
# debug platform packet-trace packet 128 fia-trace
```

Packet Tracer Statistics

- Check stats

```
# show platform packet-trace statistics
Packets Summary
  Matched  384
  Traced   129
Packets Received
  Ingress  264
```

Packet Tracer – View and Export Packet dump

- Summary View of all packets

```
# show platform packet-tracer summary
```

- Export packet dump

```
# show platform packet-tracer packet all | redirect {bootflash | tftp: | ftp:} pactrac.txt
```

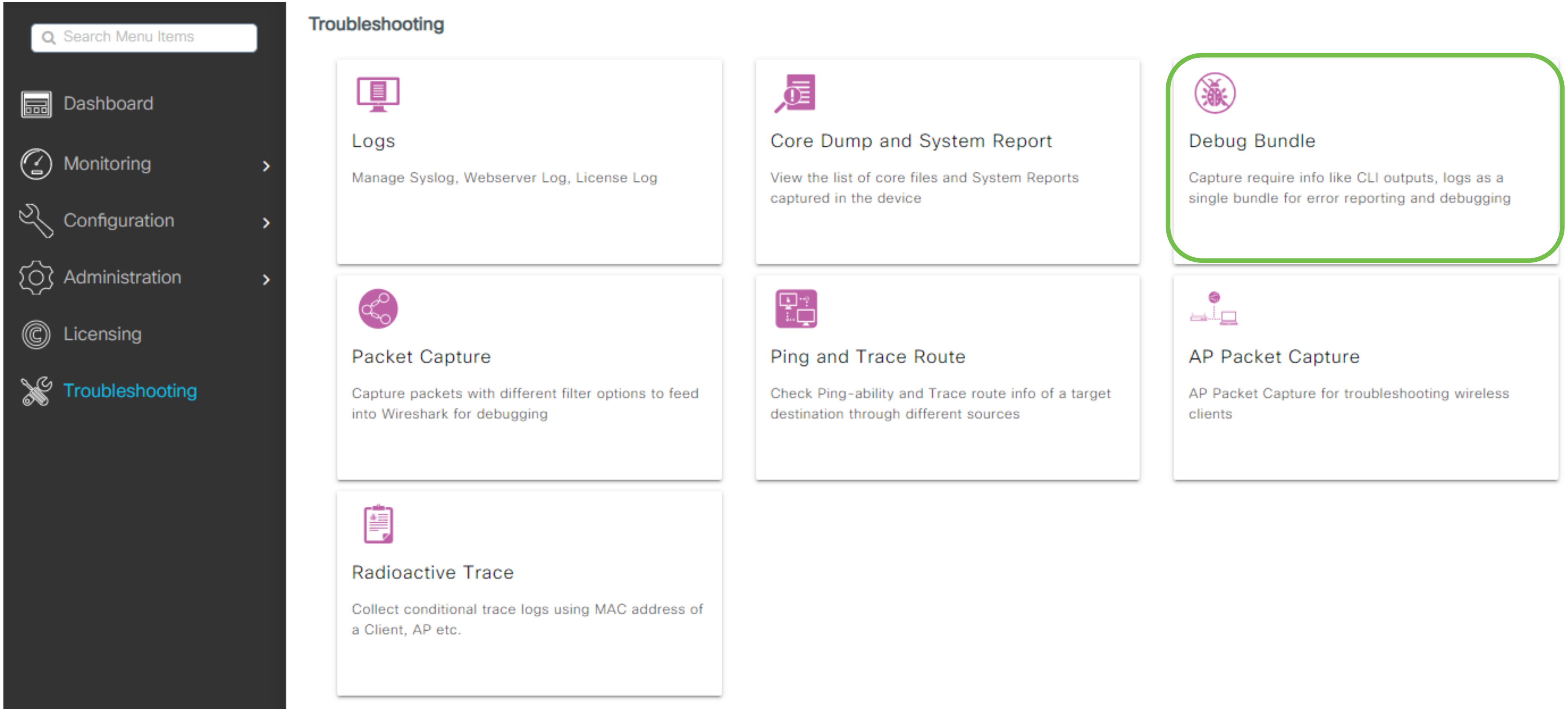
Packet Tracing – View specific packet

```
#show platform packet-trace packet 47
```

```
Feature: IPV4_INPUT_GOTO_OUTPUT_FEATUREEntry : Input - 0x8173e358
Input : Vlan1104
Output : <unknown>
Lapsed time : 4000 ns
Feature: CAPWAP_DTLS_CTRL_DECRYPT_PRE_EXT


Entry : Input - 0x8178ff90
Input : Vlan1104
Output : <unknown>
Lapsed time : 933 ns
Feature: CAPWAP_CTRL_PUNT_EXT


Entry : Output - 0x8178f660
Input : Vlan1104
Output : internal0/0/rp:0
Lapsed time : 4913 ns
```

Other Troubleshooting Tools

GUI Troubleshooting Dashboard



Troubleshooting

Logs

Manage Syslog, Webserver Log, License Log

Core Dump and System Report

View the list of core files and System Reports captured in the device

Debug Bundle

Capture require info like CLI outputs, logs as a single bundle for error reporting and debugging

Packet Capture

Capture packets with different filter options to feed into Wireshark for debugging

Ping and Trace Route

Check Ping-ability and Trace route info of a target destination through different sources

AP Packet Capture

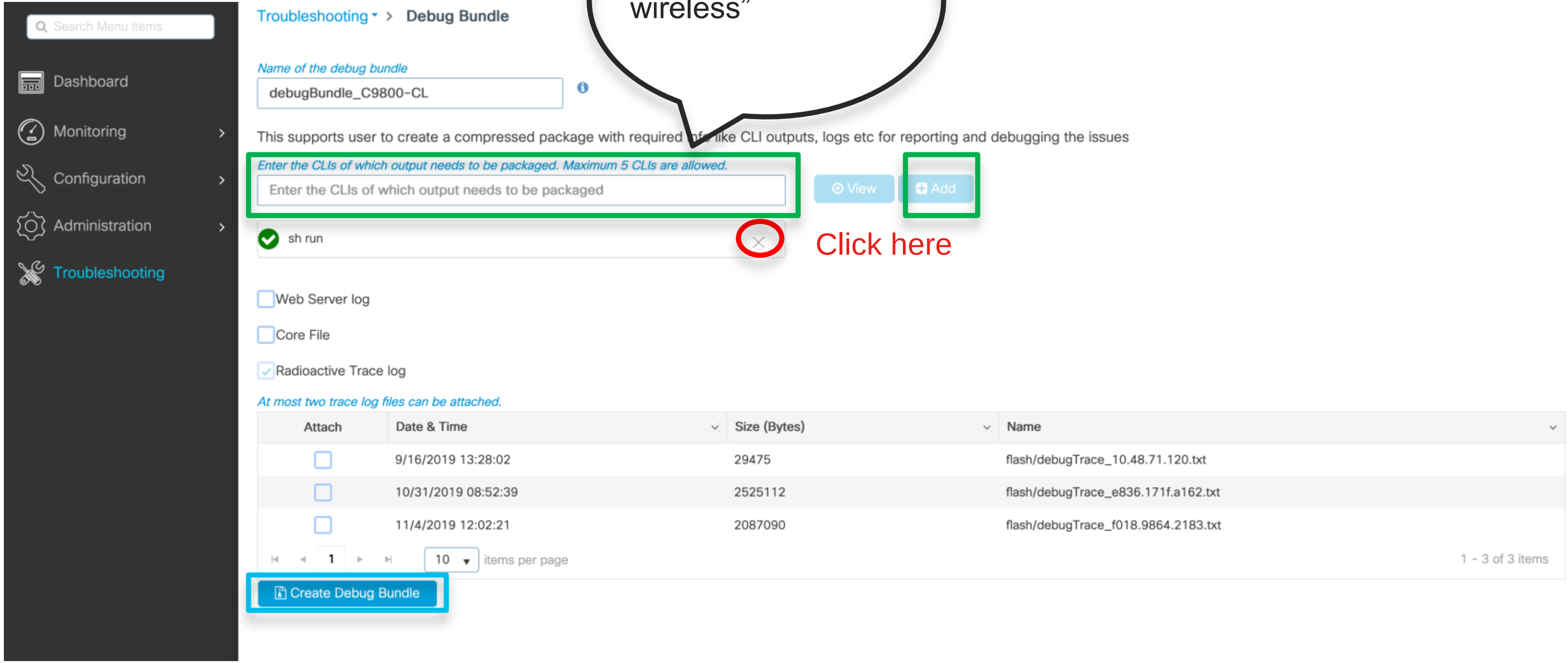
AP Packet Capture for troubleshooting wireless clients

Radioactive Trace

Collect conditional trace logs using MAC address of a Client, AP etc.

GUI Troubleshooting Dashboard

Debug Bundle Page



“show tech-support”
“show tech-support wireless”

Troubleshooting › Debug Bundle

Name of the debug bundle

debugBundle_C9800-CL

This supports user to create a compressed package with required info like CLI outputs, logs etc for reporting and debugging the issues

Enter the CLIs of which output needs to be packaged. Maximum 5 CLIs are allowed.

Enter the CLIs of which output needs to be packaged

✓ sh run

Click here

□ Web Server log

□ Core File

☑ Radioactive Trace log

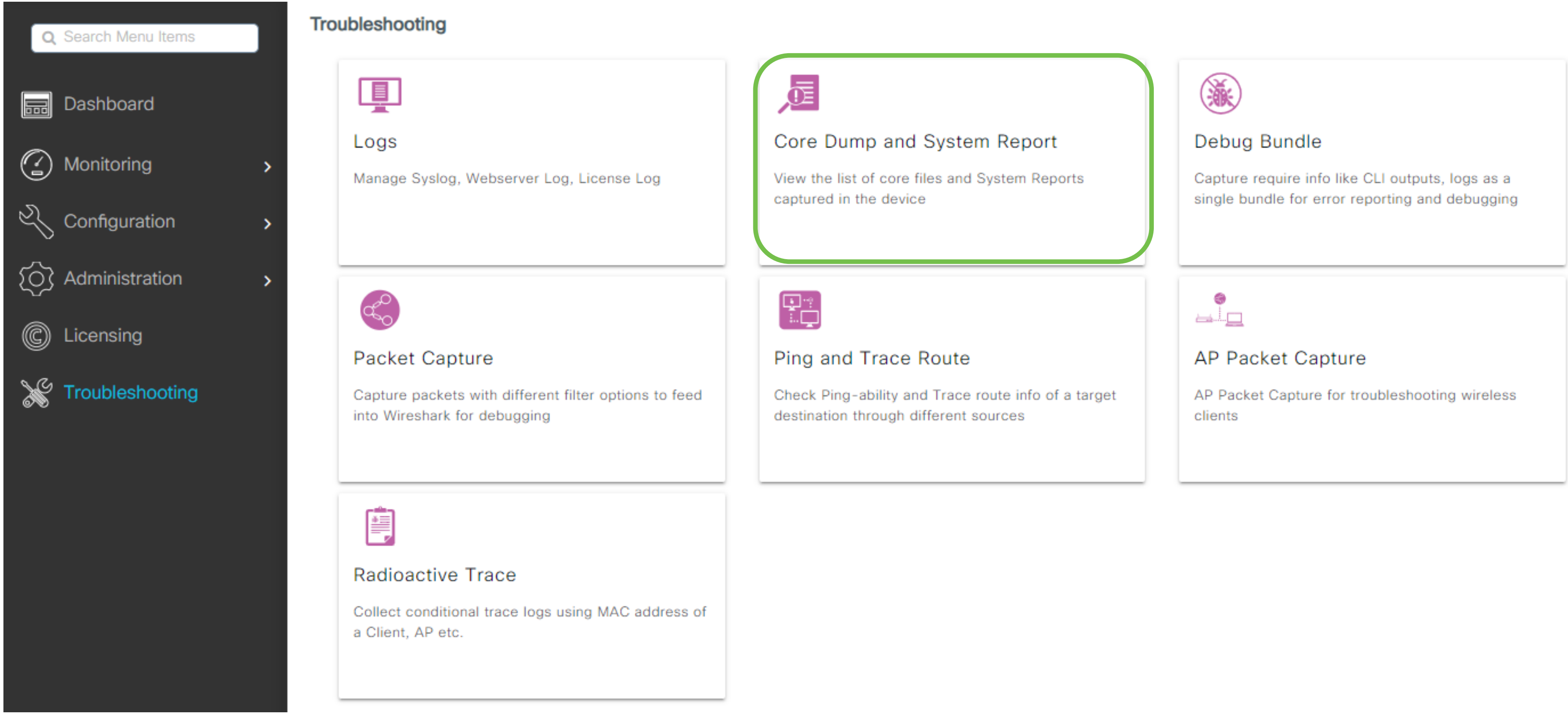
At most two trace log files can be attached.

Attach	Date & Time	Size (Bytes)	Name
□	9/16/2019 13:28:02	29475	flash/debugTrace_10.48.71.120.txt
□	10/31/2019 08:52:39	2525112	flash/debugTrace_e836.171f.a162.txt
□	11/4/2019 12:02:21	2087090	flash/debugTrace_f018.9864.2183.txt

1 | 10 items per page | 1 - 3 of 3 items

```
# show tech wireless       # show tech wireless client    #show tech wireless qos
# show tech memory.        # show tech wireless multicast #show tech wireless datapath
```

GUI Troubleshooting Dashboard



Troubleshooting

Logs

Manage Syslog, Webserver Log, License Log

Core Dump and System Report

View the list of core files and System Reports captured in the device

Debug Bundle

Capture require info like CLI outputs, logs as a single bundle for error reporting and debugging

Packet Capture

Capture packets with different filter options to feed into Wireshark for debugging

Ping and Trace Route

Check Ping-ability and Trace route info of a target destination through different sources

AP Packet Capture

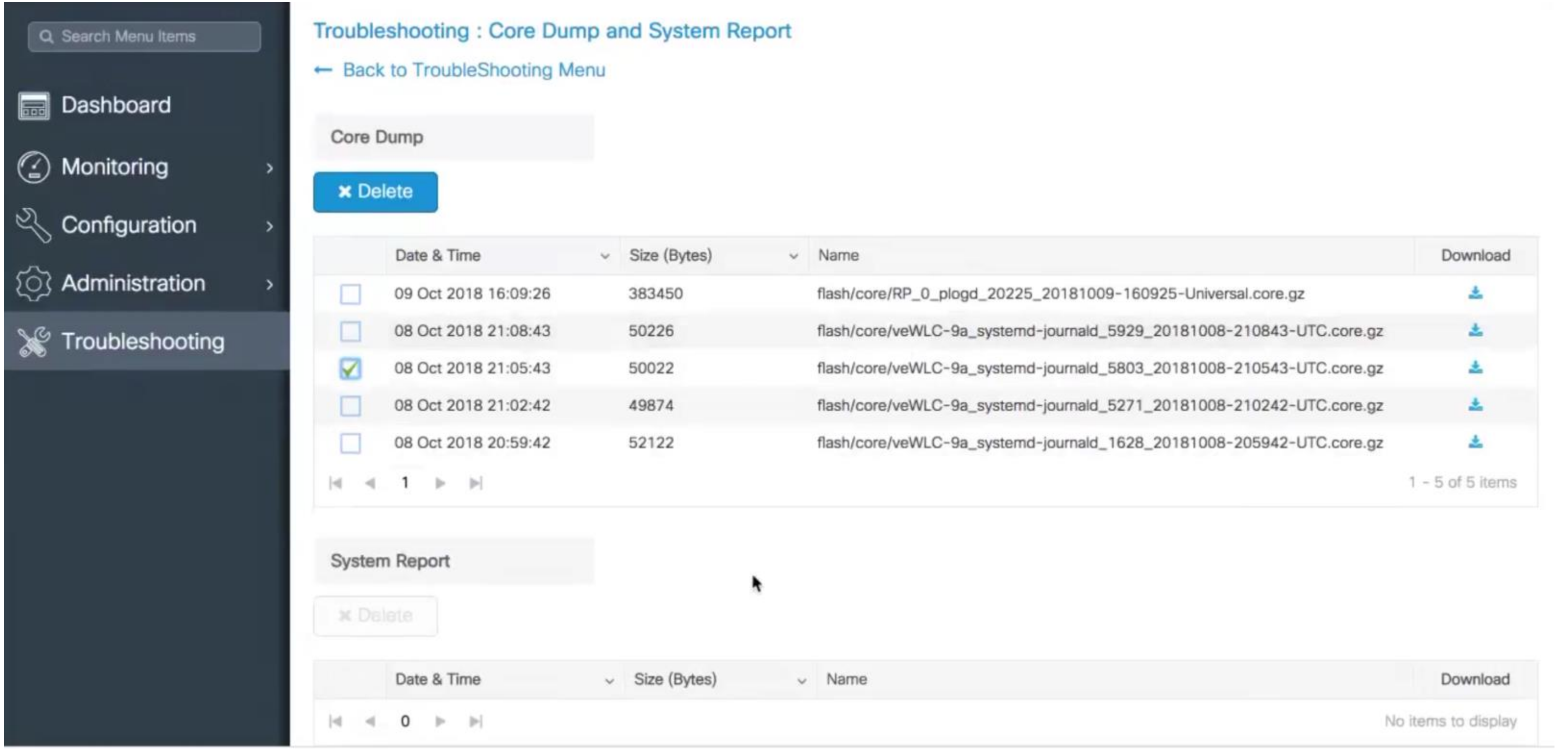
AP Packet Capture for troubleshooting wireless clients

Radioactive Trace

Collect conditional trace logs using MAC address of a Client, AP etc.

GUI Troubleshooting Dashboard

Core Dump and System Report page



Troubleshooting : Core Dump and System Report

← Back to TroubleShooting Menu

Core Dump

✖ Delete

	Date & Time	Size (Bytes)	Name	Download
□	09 Oct 2018 16:09:26	383450	flash/core/RP_0_plogd_20225_20181009-160925-Universal.core.gz	⭳
□	08 Oct 2018 21:08:43	50226	flash/core/veWLC-9a_systemd-journald_5929_20181008-210843-UTC.core.gz	⭳
☑	08 Oct 2018 21:05:43	50022	flash/core/veWLC-9a_systemd-journald_5803_20181008-210543-UTC.core.gz	⭳
□	08 Oct 2018 21:02:42	49874	flash/core/veWLC-9a_systemd-journald_5271_20181008-210242-UTC.core.gz	⭳
□	08 Oct 2018 20:59:42	52122	flash/core/veWLC-9a_systemd-journald_1628_20181008-205942-UTC.core.gz	⭳

1 | 1 - 5 of 5 items

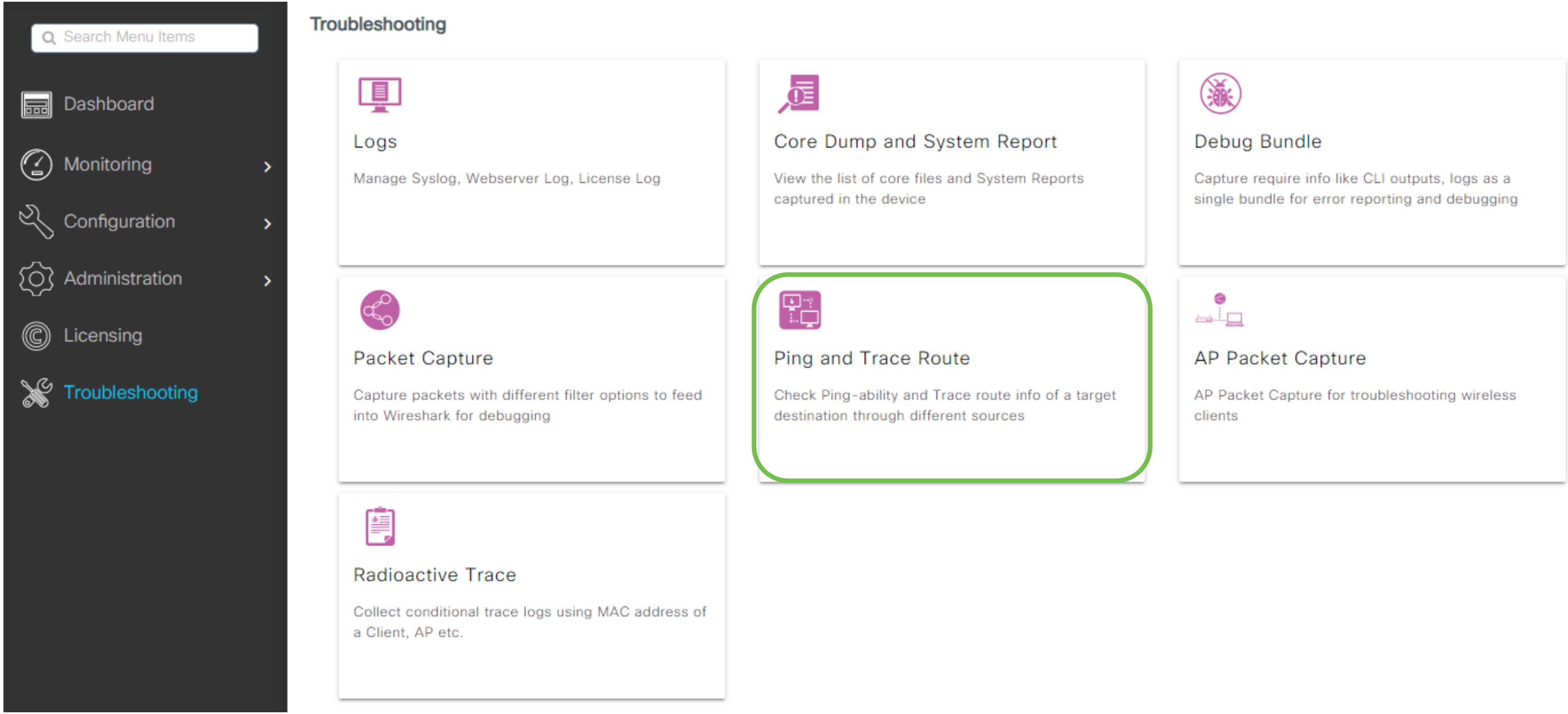
System Report

✖ Delete

	Date & Time	Size (Bytes)	Name	Download

0 | No items to display

GUI Troubleshooting Dashboard



Troubleshooting

Logs

Manage Syslog, Webserver Log, License Log

Core Dump and System Report

View the list of core files and System Reports captured in the device

Debug Bundle

Capture require info like CLI outputs, logs as a single bundle for error reporting and debugging

Packet Capture

Capture packets with different filter options to feed into Wireshark for debugging

Ping and Trace Route

Check Ping-ability and Trace route info of a target destination through different sources

AP Packet Capture

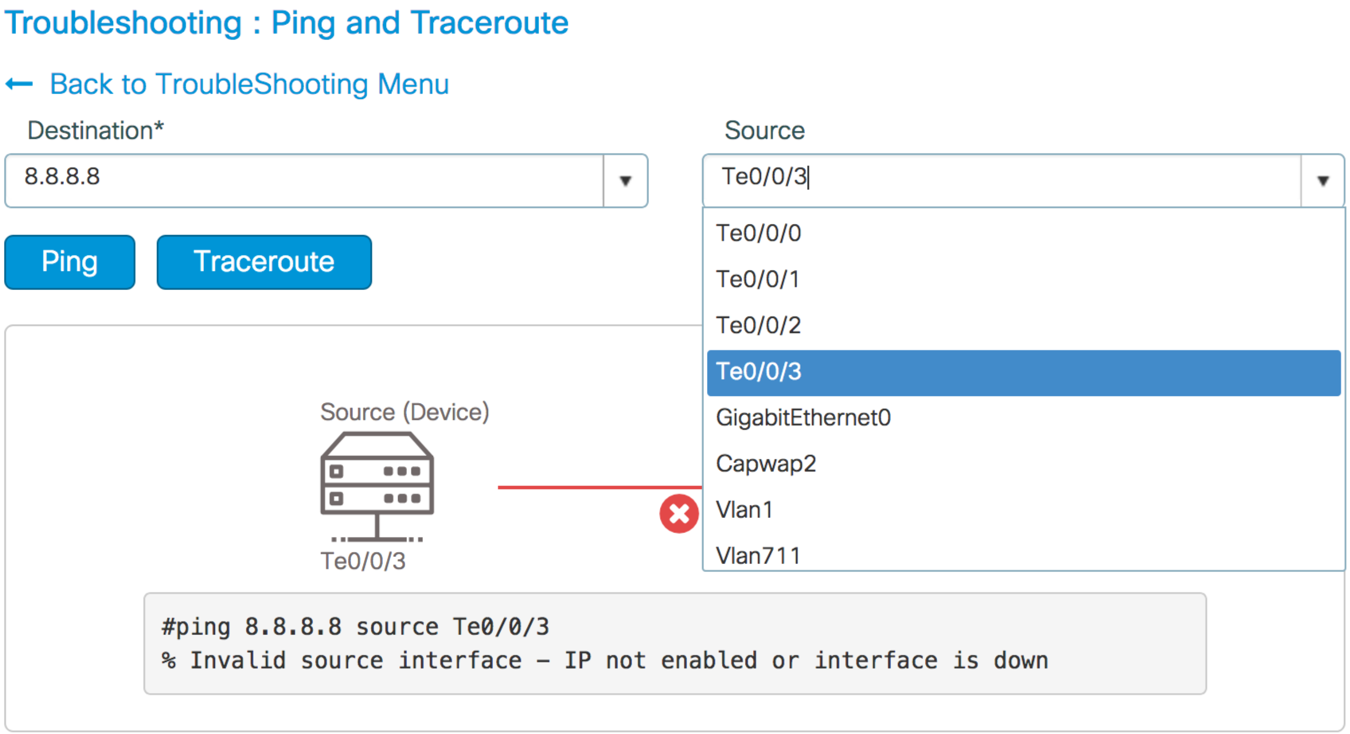
AP Packet Capture for troubleshooting wireless clients

Radioactive Trace

Collect conditional trace logs using MAC address of a Client, AP etc.

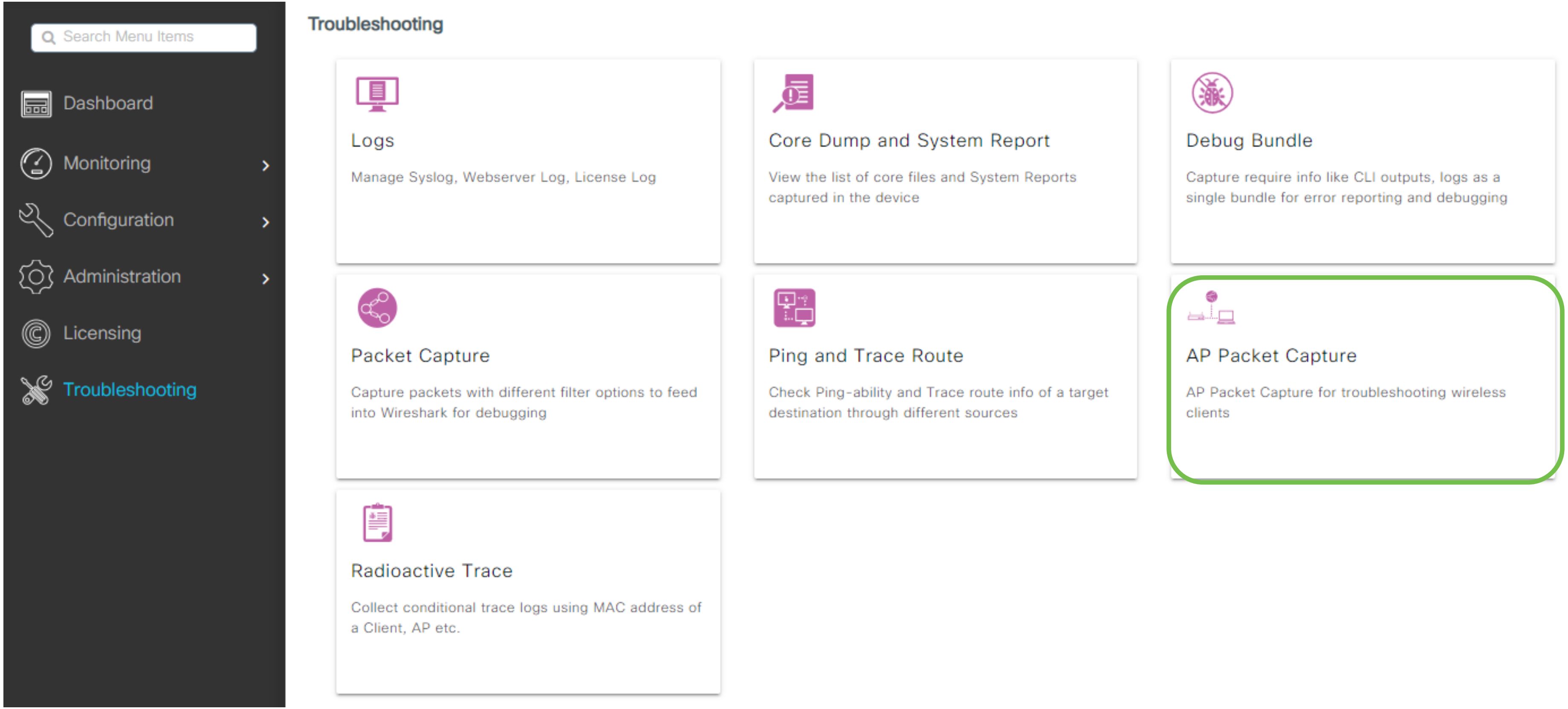
Useful commands and tools

Ping and Traceroute page



```
#ping 8.8.8.8 source Te0/0/3
% Invalid source interface – IP not enabled or interface is down
```

GUI Troubleshooting Dashboard



Troubleshooting

Logs

Manage Syslog, Webserver Log, License Log

Core Dump and System Report

View the list of core files and System Reports captured in the device

Debug Bundle

Capture require info like CLI outputs, logs as a single bundle for error reporting and debugging

Packet Capture

Capture packets with different filter options to feed into Wireshark for debugging

Ping and Trace Route

Check Ping-ability and Trace route info of a target destination through different sources

AP Packet Capture

AP Packet Capture for troubleshooting wireless clients

Radioactive Trace

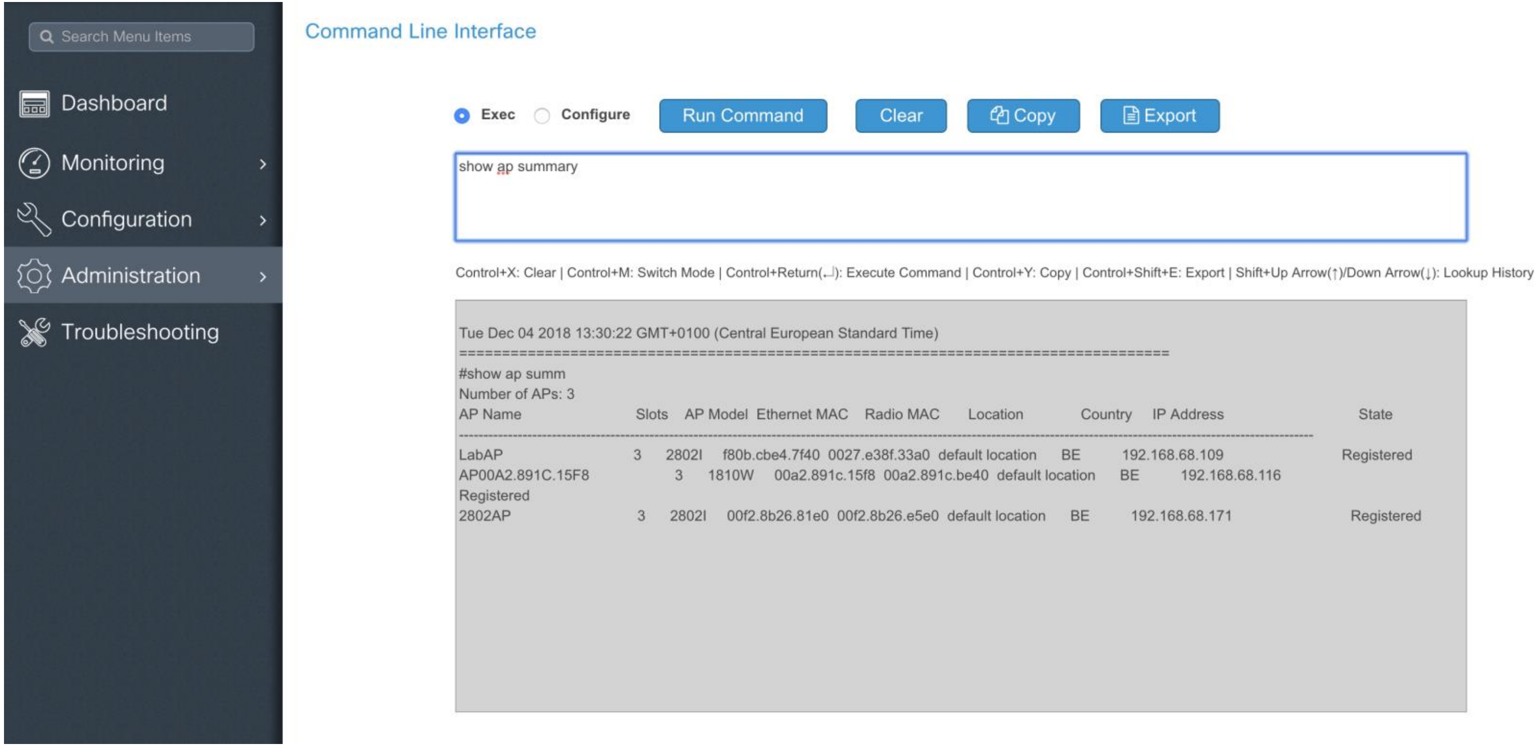
Collect conditional trace logs using MAC address of a Client, AP etc.

AP Packet Capture

- This puts AP in sniffer mode to collect over the air traces
- It is only applicable to IOS APs supported by C9800 - 1700, 2700, 3700, IW3700, AP803.
- https://www.cisco.com/c/en/us/support/docs/wireless/catalyst-9800-series-wireless-controllers/213914-configure-ap-packet-capture-on-catalyst.html

GUI based CLI Editor

Administration -> Command line interface page



● Exec ○ Configure

show ap summary

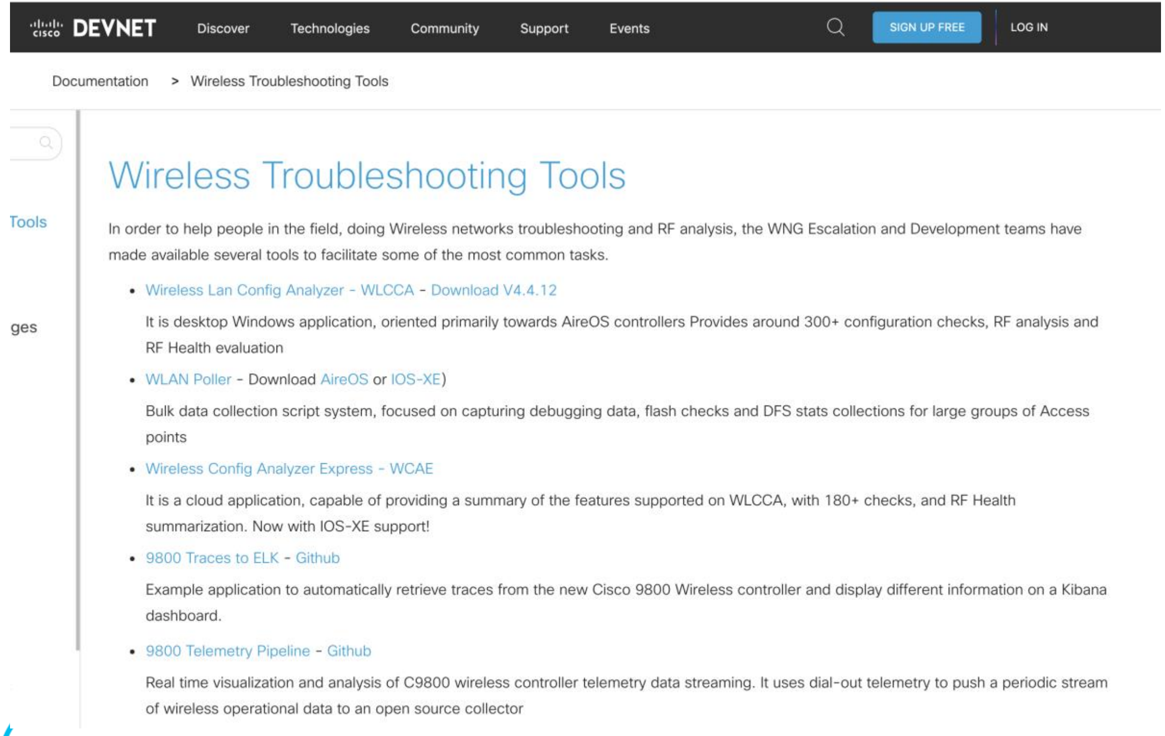
Control+X: Clear | Control+M: Switch Mode | Control+Return(↵): Execute Command | Control+Y: Copy | Control+Shift+E: Export | Shift+Up Arrow(↑)/Down Arrow(↓): Lookup History

```
Tue Dec 04 2018 13:30:22 GMT+0100 (Central European Standard Time)
================================================================================
#show ap summ
Number of APs: 3
AP Name                  Slots   AP Model  Ethernet MAC     Radio MAC       Location          Country     IP Address                    State
-----------------------------------------------------------------------------------------------------------------------------------------------
LabAP                      3      2802I     f80b.cbe4.7f40  0027.e38f.33a0  default location      BE         192.168.68.109                  Registered
AP00A2.891C.15F8                3      1810W     00a2.891c.15f8  00a2.891c.be40  default location      BE         192.168.68.116
Registered
2802AP                     3      2802I     00f2.8b26.81e0  00f2.8b26.e5e0  default location      BE         192.168.68.171                  Registered
```

Other Tools

Wireless Troubleshooting Tools

https://developer.cisco.com/docs/wireless-troubleshooting-tools/



Wireless Troubleshooting Tools

In order to help people in the field, doing Wireless networks troubleshooting and RF analysis, the WNG Escalation and Development teams have made available several tools to facilitate some of the most common tasks.

- Wireless Lan Config Analyzer - WLCCA - Download V4.4.12

  It is desktop Windows application, oriented primarily towards AireOS controllers Provides around 300+ configuration checks, RF analysis and RF Health evaluation

- WLAN Poller - Download AireOS or IOS-XE)

  Bulk data collection script system, focused on capturing debugging data, flash checks and DFS stats collections for large groups of Access points

- Wireless Config Analyzer Express - WCAE

  It is a cloud application, capable of providing a summary of the features supported on WLCCA, with 180+ checks, and RF Health summarization. Now with IOS-XE support!

- 9800 Traces to ELK - Github

  Example application to automatically retrieve traces from the new Cisco 9800 Wireless controller and display different information on a Kibana dashboard.

- 9800 Telemetry Pipeline - Github

  Real time visualization and analysis of C9800 wireless controller telemetry data streaming. It uses dial-out telemetry to push a periodic stream of wireless operational data to an open source collector

Agenda

Chapters

- Hardware and Software Architecture
- Life of a Packet

1

- New Config Model
- Deployment Considerations

2

- GUI Troubleshooting Dashboard
- IOS-XE Tracing, Packet Capture & Packet Tracer

3

- Health and KPI Monitoring
- Conclusion

4

HW monitoring

HW sensors and status

```
# show environment all
```

```
Sensor List:  Environmental Monitoring
 Sensor           Location          State             Reading
 Vin              P0                Normal            119 V AC
 Iin              P0                Normal            2 A
 Vout             P0                Normal            12 V DC
 Iout             P0                Normal            20 A
 Temp1            P0                Normal            33 Celsius
 Temp2            P0                Normal            29 Celsius
 Temp3            P0                Normal            37 Celsius
 VRRX1: VX1       R0                Normal            751 mV
 VRRX1: VX2       R0                Normal            6909 mV
 VRRX1: VX3       R0                Normal            1216 mV
```

Virtual “HW” monitoring

Box specifications and environment

```
#sh platform software system all

Processor Details
=================
Number of Processors : 4
Processor : 1 - 4
vendor_id : GenuineIntel
cpu MHz   : 2266.747
cache size : 8192 KB
Crypto Supported : No
model name : Int

Hypervisor Details
===================
Hypervisor: VMWARE
Manufacturer: VMware, Inc.
Product Name: VMware Virtual Platform
```

AP Health

Verifying AP discovery

# show wireless stats ap discovery

```
Discovery requests received from total number of APs : 3

AP Radio MAC    AP Ethernet MAC  IP Address                        Last Success time       Last failure type           Last failure time
--------------------------------------------------------------------------------------------------------------------------------------------

0062.ecaa.de80  0042.68a0.ee78   192.168.26.101                      05/28/19 10:00:02       None                      NA
00a3.8ec2.da00  002c.c899.b9ac   192.168.25.102                     05/28/19 10:00:02       None                      NA
cc16.7e30.3980  58ac.78de.891e   192.168.26.102                     05/28/19 10:00:09       Non-wireless Mgmt interface   NA
```

- Single view for all Aps that tried to find the controller

AP Health

AP reliability

# show ap uptime

```
Number of APs: 3

AP Name                    Ethernet MAC    Radio MAC       AP Up Time                       Association Up Time
------------------------------------------------------------------------------------------------------------------------
ap3800i-r2-sw1-te0-1       0042.68a0.ee78  0062.ecaa.de80  1 day 0 hour 37 minutes  1 day 0 hour 21
ap2800-r2-sw1-2-0-4        002c.c899.b9ac  00a3.8ec2.da00  1 day 0 hour 38 minutes  1 day 0 hour 21
ap3800i-r2-sw1-te0-2       58ac.78de.891e  cc16.7e30.3980  1 day 0 hour 36 minutes  1 day 0 hour 21
```

- Single view:
  - AP crashes
  - CAPWAP bounces

AP Health

Verifying AP join

# show wireless stats ap join summary

```
Number of APs: 2

Base MAC       Ethernet MAC    AP Name                   IP Address      Status              Last Failure Type    Last Disconnect Reason
--------------------------------------------------------------------------------------------------------------------------------------------

0062.ec06.8d10  0000.0000.0000  NA                                        NA              Not Joined        Dtls                  NA
00be.75ba.1220  0000.0000.0000  NA                                        NA             Not Joined  Dtls                         NA
7c0e.cea0.7680  58f3.9cc4.4864  AP58f3.9cc4.4864      192.168.16.92      Not Joined                       NA                    Heart beat timer expiry
84b8.021d.1c70  64f6.9d58.5d3c  2702I-sniffer        192.168.16.198     Joined      Join                   Wtp reset config cmd sent
a80c.0ddb.c720  a80c.0dd2.1fa8  APa80c.0dd2.1fa8                          192.168.18.52      Joined      NA                         DTLS alert from AP
```

- Single view:
    - AP Join failures
    - Reason codes
    - AP mac/IP for debugging

AP Health

Verifying DTLS

# show wireless dtls connections

```
AP Name              Local Port    Peer IP      Peer Port  Version     Ciphersuite
--------------------------------------------------------------------------------------------
APD4E8.8019.49E0        Capwap_Ctrl   170.85.125.43  5250      DTLSv1.0   TLS_NUM_RSA_WITH_AES_128_CBC_SHA
EDU_BR_01_00_01_1852     Capwap_Ctrl   170.85.142.18  5264      DTLSv1.0   TLS_NUM_RSA_WITH_AES_128_CBC_SHA
EDU_BR_01_00_02_3702     Capwap_Ctrl   170.85.125.14  56998     DTLSv1.0   TLS_NUM_RSA_WITH_AES_128_CBC_SHA
EDU_BR_01_00_03_1832     Capwap_Ctrl   170.85.145.85  5264      DTLSv1.0   TLS_NUM_RSA_WITH_AES_128_CBC_SHA
EDU_BR_01_00_10_1832     Capwap_Ctrl   170.85.151.11  5272      DTLSv1.0   TLS_NUM_RSA_WITH_AES_128_CBC_SHA
EDU_BR_01_00_13_3702     Capwap_Ctrl   170.85.125.20  62903     DTLSv1.0   TLS_NUM_RSA_WITH_AES_128_CBC_SHA
```

- Single view:
  - Connections per AP
  - Ciphers in use
  - Source ports for NAT/PAT problems
  - Mobility will show here

AP Health

What happened

# show wireless stats ap history

```
AP Name                  Ethernet MAC   Event     Time          Recent Disconnect Time   Disconnect Reason        Disconnect Count
---------------------------------------------------------------------------------------------------------------------------------
ap2800-r2-sw1-2-0-4          002c.c899.b9ac  Joined     05/29/19 10:49:35  NA
ap2800-r2-sw1-2-0-4          002c.c899.b9ac  Disjoined  05/29/19 10:48:18  NA                 Heart beat timer expiry
ap2800-r2-sw1-2-0-4          002c.c899.b9ac  Joined     05/28/19 10:00:12  NA
ap3800i-r2-sw1-te0-1         0042.68a0.ee78  Joined     05/28/19 10:00:13  NA
ap3800i-r2-sw1-te0-2         58ac.78de.891e  Joined     05/28/19 10:00:19  NA
```

- Single view:
  - Recent events per AP
  - What happened and when
  - No debug or data collection needed

AP Health

Verifying AP Plumbed Path

# show ap summary

```
Number of APs: 1

AP Name              Slots    AP Model  Ethernet MAC    Radio MAC      Location            Country    IP Address          State

------------------------------------------------------------------------------------------------------------------------------AP4C77.6D9E.6162       3      4800      4c77.6d9e.6162  7069.5a51.4ec0  default
location       BE        192.168.79.249       Registered
```

# show platform software capwap chassis active R0

```
sh platform software capwap chassis active R0

 Tunnel ID  AP MAC        Type          IP             Port

------------------------------------------------------------------

0x90000004  7069.5a51.4ec0  Data          192.168.79.249    5272

0xa0000001  0000.0000.0000  Mobility Data   10.48.71.113     16667
```

AP Health

Verifying AP Plumbed Path

```
# show platform software capwap chassis active F0

Tunnel ID  AP MAC          Type           IP              Port      AOM ID  Status
--------------------------------------------------------------------------------------
0x90000004  7069.5a51.4ec0  Data          192.168.79.249    5272       567  Done
0xa0000001  0000.0000.0000  Mobility Data   10.48.71.113     16667       519  Done
```

```
# show platform hardware chassis active qfp feature wireless capwap cpp-client summary

cpp_if_hdl     pal_if_hdl       AP MAC           Src IP          Dst IP       Dst Port  Tun Type
----------------------------------------------------------------------------------------------
    0X33       0XA0000001   0000.0000.0000    10.48.39.30      10.48.71.113     16667    MOBILITY
    0X34       0X90000004   7069.5a51.4ec0    10.48.39.30     192.168.79.249     5272      DATA
```

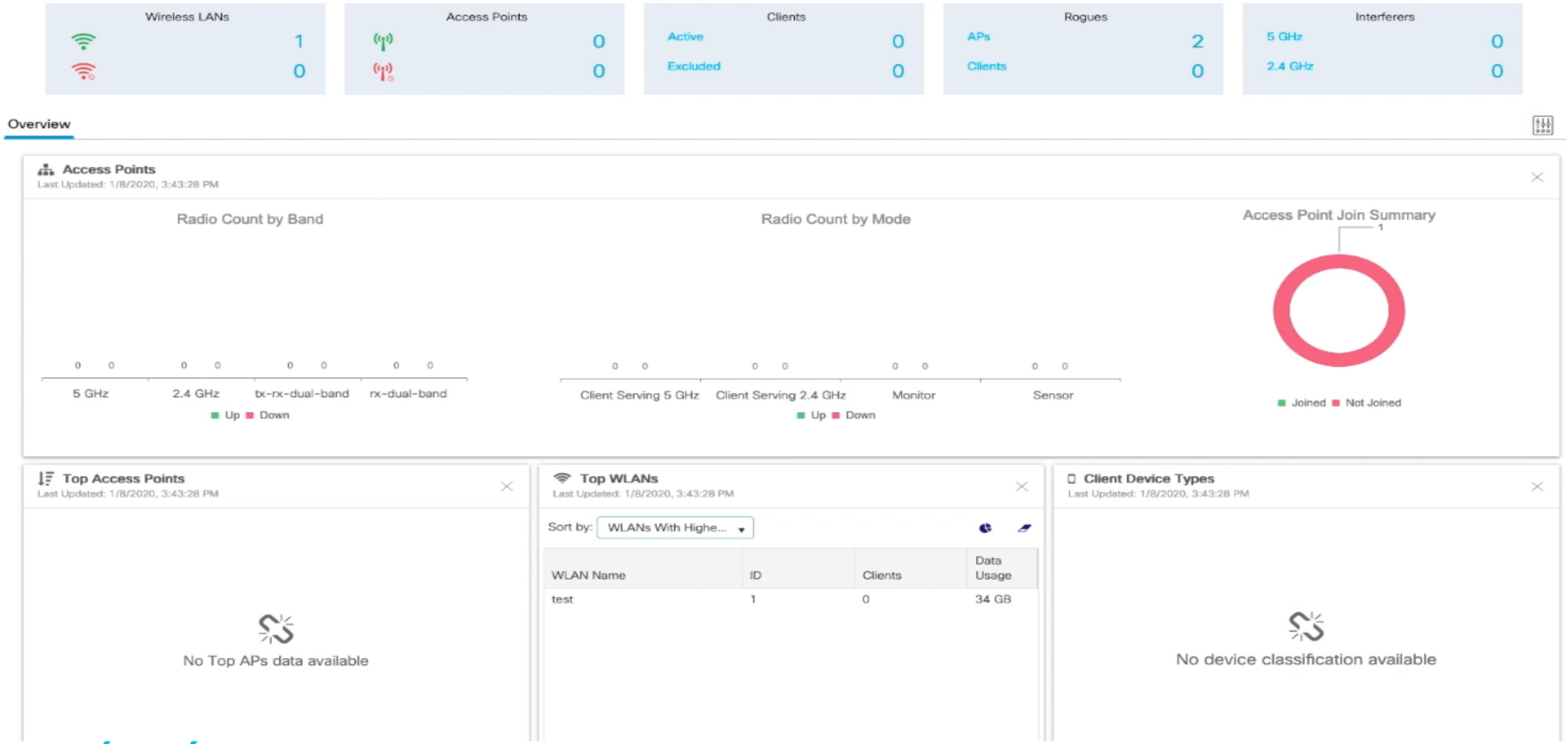
AP Health

Verifying AP Plumbed Path

# show platform hardware chassis active qfp feature wireless capwap datapath summary

```
Vrf Src Port Dst IP         Dst Port Input Uidb Output Uidb Instance Id

--- -------- ------         -------- ---------- ----------- -----------

0   5247     192.168.79.249  5272     65490      65484       3

0   16667    10.48.71.113    16667    65491      65485       0
```

Troubleshooting APs the easy way



Wireless LANs		Access Points		Clients		Rogues		Interferers	
	1		0	Active	0	APs	2	5 GHz	0
	0		0	Excluded	0	Clients	0	2.4 GHz	0

Overview

Access Points
Last Updated: 1/8/2020, 3:43:28 PM

Top Access Points
Last Updated: 1/8/2020, 3:43:28 PM

No Top APs data available

Top WLANs
Last Updated: 1/8/2020, 3:43:28 PM

Sort by: WLANs With Highe...

WLAN Name	ID	Clients	Data Usage
test	1	0	34 GB

Client Device Types
Last Updated: 1/8/2020, 3:43:28 PM

No device classification available

Client Health Monitoring

The SUPER command

# show wireless stats client detail

```
Total Number of Clients : 4
Protocol Statistics
-----------------------------------------------------------------------------
Protocol                          Client Count
802.11b                                0
802.11g                                0
802.11a                                0
802.11n-2.4 GHz                        0
802.11n-5 GHz                          0
802.11ac                               4
802.11ax-5 GHz                         0
802.11ax-2.4 GHz                       0

Client Summary
-----------------------------
Current Clients      : 4
Excluded Clients     : 1
Disabled Clients     : 0
Foreign Clients      : 0
Anchor Clients       : 0
Local Clients        : 4
```

- Single view:
  - Total clients connected
  - Per Protocol distribution
  - State Distribution : easy to spot network wide problems

Client Health Monitoring

The SUPER command (part 2)

```
client global statistics:
--------------------------------------------------------------------------------
Total association requests received                  : 22280
Total association attempts                           : 21381
Total FT/LocalAuth requests                          : 0
Total association failures                           : 1
…
Total AID allocation failures                        : 0
Total AID free failures                              : 0
Total roam attempts                                  : 13435
  Total CCKM roam attempts                           : 0
  Total 11r roam attempts                            : 5454
…
Total add mobiles sent                               : 33024
Total delete mobiles sent                            : 16664
…
Total key exchange attempts                          : 7414
Total broadcast key exchange attempts                : 14298
Total broadcast key exchange failures                : 0
Total eapol key sent                                 : 35720
Total eapol key received                             : 27565
…
```

- Single view:
  - 98 different stats counters
  - Easy to spot:
    - Frequent Bcast rotation issues
    - Frequent L2/L3 auth failures
    - Frequent IP address learning failures
  - Roaming types

Client Health Monitoring

The SUPER command (part 3)

```
client state statistics:
-----------------------------------------------------------------------------
Average Time in Each State (ms)
  Associated State    : 0
  L2 State       : 85
  Mobility State : 2
  IP Learn State : 2117
  L3 Auth State   : 0

Average Run State Latency (ms) : 1102

Average Run State Latency without user delay (ms) : 1061

Latency Distribution (ms)
  1 - 100    : 278025
  100 - 200  : 11511
  200 - 300  : 5590
  300 - 600  : 3519
  600 - 1000 : 6546
  1000+      : 41184
```

- Single view:
  - Average time per state
  - Spotting performance problems
  - Variations over time

Client Health Monitoring

The SUPER command (part 4)

```
Webauth HTTP Statistics
-----------------------
  Intercepted HTTP requests   : 0
  IO Read events              : 0
  Received HTTP messages      : 0
…
Time spent in each httpd states (in msecs)
                             Total      Max        Min        Samples
----------------------------------------------------------------------
IO Reading state                0          0          0          0
IO Writing state                0          0          0          0
IO AAA state                    0          0          0          0
Method after reading            0          0          0          0
…
Webauth HTTP status counts
--------------------------
 HTTP 200 OK                    : 0
 HTTP 201 Created               : 0
 HTTP 202 Accepted              : 0
 HTTP 203 Provisional Info      : 0
```

Single view:

- Webauth HTTP statistics
- Webauth HTTP response codes

Client Health Monitoring

The SUPER command (part 5)

```
Webauth backpressure queue counters
-----------------------------------
Pending SSL handshakes              : 0
Pending HTTPS new requests          : 0
Pending AAA replies                 : 0

Dot1x Global Statistics
-----------------------
RxStart = 97 RxLogoff = 0  RxResp = 1095 RxRespID = 282
RxReq = 0 RxInvalid = 0  RxLenErr = 0
RxTotal = 1486
TxStart = 0 TxLogoff = 0  TxResp = 0
TxReq = 1679 ReTxReq = 362  ReTxReqFail = 64
TxReqID = 643 ReTxReqID = 228  ReTxReqIDFail = 3
TxTotal = 2322
```

Single view:

- Webauth queue full issues
- SSL session exhaustion
- Dot1x statistics

Client Health Monitoring

The SUPER command (part 6)

```
Total client delete reasons
---------------------------
Controller deletes
---------------------------
---------------------------
No Operation                                  : 0
Unknown                                       : 0
Session Manager                               : 0
Connection timeout                            : 0
Datapath plumb                                : 0
….
---------------------------
Informational Delete Reason
----------------------------
Mobility WLAN down                            : 0
AP upgrade                                    : 0
L3 authentication failure                     : 0
AP down/disjoin                               : 0
MAC authentication failure                    : 0
…..
```

Single view:

- Client delete reasons categorized by
  - •Controller initiated delete
  - •AP initiated delete
  - •Network wide problem isolation

Client Health Monitoring

The SUPER command (part 6) continued

```
-----------------------------
Client initiate delete
-----------------------------
Deauthentication or disassociation request                 : 0
Client DHCP                                              : 0
Client EAP timeout                                       : 0
Client 8021x failure                                     : 0
Client device idle                                       : 0
Client captive portal security failure                   : 0
…
-----------------------------
AP Deletes
-----------------------------
AP initiated delete when client is sending disassociation    : 0
AP initiated delete for idle timeout                     : 0
AP initiated delete for client ACL mismatch                 : 0
AP initiated delete for AP auth stop                     : 0
AP initiated delete for association expired at AP          : 0
AP initiated delete for 4-way handshake failed             : 0.
```

Single view:

- Client initiated delete reasons
- AP initiated delete reasons

Client Health

Verifying Client Plumbed Path

# show wireless client summary

```
Number of Clients: 1

MAC Address    AP Name                              Type ID   State          Protocol Method     Role

------------------------------------------------------------------------------------------------------------------------

ccc0.796d.7ca0 sudha-9115                           WLAN 1    Run            11ac     None       Local
```

FMAN-RP view # show platform software wireless-client chassis active R0

```
 ID  MAC Address      WLAN  Client State

---------------------------------------------------

0xa0000001  ccc0.796d.7ca0     1  Run
```

Client Health

Verifying Client Plumbed Path

FMAN-FP view # show platform software wireless-client chassis active F0

```
 ID  MAC Address      WLAN  Client State         AOM ID  Status
------------------------------------------------------------------------------------------
0xa0000001  ccc0.796d.7ca0     1  Run                    480  Done
```

CPP-Client view

# show platform hardware chassis active qfp feature wireless wlclient cpp-client summary

```
CPP IF_H     DPIDX      MAC Address    VLAN  CT  MCVL  AS  MS  E            WLAN              POA
----------------------------------------------------------------------------------------------------------
   0X30    0XA0000001  ccc0.796d.7ca0  1104  RG   0    RN  LC  N          clus-dot1x          0x90000004
```

Client Health

Verifying Client Plumbed Path

CPP Dataplane view

```
# show platform hardware chassis active qfp feature wireless wlclient datapath summary

Vlan   pal_if_hdl   mac            Input Uidb Output Uidb
------ ------------ -------------- ---------- -----------
1104   0xa0000001   ccc0.796d.7ca0 95954      95952
```

CPU Health

One CPU command to view Control and Data Plane

C9800-40#show processes cpu platform sorted | inc CPU|Core|Pid|wncd

```
CPU utilization for five seconds:  1%, one minute:  0%, five minutes:  0%
Core 0: CPU utilization for five seconds:  0%, one minute:  1%, five minutes:  0%
Core 1: CPU utilization for five seconds:  0%, one minute:  5%, five minutes:  1%
Core 2: CPU utilization for five seconds:  0%, one minute:  1%, five minutes:  0%
Core 3: CPU utilization for five seconds:  1%, one minute:  1%, five minutes:  0%
Core 4: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 5: CPU utilization for five seconds: 18%, one minute:  2%, five minutes:  1%
Core 6: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  1%
Core 7: CPU utilization for five seconds:  0%, one minute:  1%, five minutes:  1%
Core 8: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 9: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 10: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 11: CPU utilization for five seconds:  1%, one minute:  1%, five minutes:  1%
Core 12: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 13: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 14: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
Core 15: CPU utilization for five seconds:  0%, one minute:  0%, five minutes:  0%
   Pid    PPid    5Sec    1Min    5Min  Status        Size  Name
 28464   27442      0%      0%      0%  S           230876  wncd_4
 28068   26892      0%      0%      0%  S           232820  wncd_3
 27604   26264      0%      0%      0%  S           232480  wncd_2
 27131   25714      0%      0%      0%  S           232232  wncd_1
 26538   25089      0%      0%      0%  S           340352  wncd_0
```

Memory Health

Usage Thresholds and Periodic Stats

#show platform resources

```
**State Acronym: H - Healthy, W - Warning, C - Critical
Resource              Usage             Max          Warning       Critical      State
-------------------------------------------------------------------------------------------------
RP0 (ok, active)                                                              H
Control Processor   0.49%             100%          80%          90%          H
DRAM               3689MB(11%)        31703MB      88%          93%          H
harddisk            0MB(0%)           0MB          90%          95%          H
```

#show processes memory platform accounting

```
Hourly Stats

process            callsite_ID(bytes)     max_diff_bytes   callsite_ID(calls)  max_diff_calls   tracekey                                  timestamp(UTC)
------------------------------------------------------------------------------------------------------------------------------
smand_rp_0          1478252547        2869451        1478252548        116         1#fc449c9a426b026ec2d2fd46be141029      2020-08-27 17:04
keyman_rp_0          1617978370        1634978        1617978370        4769        1#b562f2fb8268b9d2026fca73e3894925      2020-08-04 21:51
nginx_rp_0          1615492096        1048576        1615492097        201         1#7f3039c9ee2986658bab6fcd69068dbd      2020-08-27 20:38
```

Memory Health

Per process usage sorted highest to lowest

#show processes memory platform sorted

```
System memory: 32464768K total, 3777808K used, 28686960K free,
Lowest: 28660184K
   Pid    Text      Data   Stack   Dynamic      RSS           Name
--------------------------------------------------------------------
 10927  342655   1288060     136       364  1288060   linux_iosd-imag
 26538     850    340736     136      8944   340736         wncd_0
 25701     147    295724    3952      6044   295724         wncmgrd
  1884     253    240256     136     41772   240256           dbm
 28068     850    233224     136      8620   233224         wncd_3
 27604     850    232904     136      8620   232904         wncd_2
 27131     850    232560     136      8620   232560         wncd_1
 24961   15020    231420     136     30144   231420   fman_fp_image
 28464     850    231320     136      8620   231320         wncd_4
 27112      94    188496     136     35956   188496      cpp_cp_svr
  5449      83    167564     136      3148   167564          pubd
  2171      63    165992     136       116   165992      cli_agent
 28806      63    162212     136      4012   162212           rrm
 29386      61    153400     136      3256   153400         rogued
 31206     178    147692     136      5256   147692        sessmgrd
 30069     928    146180     136      3172   146180          nmspd
  ………
```

Data Plane Health

Overall Utilization

#show platform hardware chassis active qfp datapath utilization

```
CPP 0: Subdev 0                 5 secs        1 min         5 min         60 min
Input:  Priority (pps)           2            2             2             2
             (bps)              1184          2480          2704          2720
  Non-Priority (pps)            18            14            15            16
             (bps)              11832         11304         12688         14632
        Total (pps)             20            16            17            18
             (bps)              13016         13784         15392         17352
Output: Priority (pps)          0             0             0             0
             (bps)              0             0             0             0
  Non-Priority (pps)            17            7             8             9
             (bps)              19712         14256         15024         30480
        Total (pps)             17            7             8              9
             (bps)              19712         14256         15024         30480
Processing: Load (pct)     0                  0             0             0
```

Data Plane Health

Global Drop Statistics

#show platform hardware chassis active qfp statistics drop all | inc Global|Wls

```
Global Drop Stats                Packets          Octets
PuntGlobalPolicerDrops                           0            0
SdwanGlobalDrop                                  0            0
WlsCapwapError                   117162         10562887
WlsCapwapFragmentationErr                        0            0
WlsCapwapNoUidb                                  0            0
WlsCapwapReassAllocErr                           0            0
WlsCapwapReassFragConsume                1083          1483710
WlsCapwapReassFragDrop                  0            0
WlsClientError                   1           94
WlsClientFNFV9Err                                0            0
WlsClientFNFV9Report                             0            0
WlsDtlsProcessingError                           0            0
```

Data Plane Health

Access Point Drop Statistics

#show platform hardware chassis active qfp feature wireless capwap datapath statistics drop all

```
Drop Cause                                                      Packets         Octets
================================================================================ ====================
Wls Capwap unsupported link type Error                          0           0
Wls Capwap invalid tunnel Error                                 0           0
Wls Capwap input config missing Error                           0           0
Wls Capwap invalid TPID Error                                   0           0
Wls Capwap ingress parsing Error                                0           0
Wls Capwap ipv4 tunnel not found Error                          99         27205
Wls Capwap ipv6 tunnel not found Error                          0           0
Wls Capwap tunnel header add Error                              0           0
Wls Capwap mobility tunnel header add Error                     0           0
Wls Capwap ingress dot3 ingress processing Error                                0           0
Wls Capwap tunnel ingress unsufficient packet data                              0           0
Wls Capwap tunnel ingress capwap hlen Error                     0           0
Wls Capwap ingress fragment capwap payload length Error                         0           0
Wls Capwap ingress non-frag capwap payload length Error             0           0
Wls Capwap ingress dot11_4 snap header len Error                                0           0
Wls Capwap ingress dot11_4 Invalid SNAP header                                  0           0
Wls Capwap ingress dot11 ingress dot11_fc Error                                 0           0
Wls Capwap ingress dot11 ingress processing Error                               0           0
Wls Capwap invalid DTLS header length Error                     0           0
Wls Capwap invalid Capwap header type Error                     0           0
```

Data Plane Health

Client Drop Statistics

# show platform hardware chassis active qfp feature wireless wlclient datapath statistics drop all

```
Drop Cause                                                  Packets          Octets
============================================================ ==================== ====================
Wls Client V6 Max Address Error                             0                0
Wls Client IPGlean Counter Index Error                      0                0
Wls Client IPGlean Counter Unchanged Error                  0                0
Wls Client IPGlean alloc no memory Error                    0                0
Wls Client iplearn l2 punt data packet skip                 0                0
Wls Client iplearn v4 punt data packet skip                 0                0
Wls Client iplearn v6 punt data packet skip                 0                0
Wls Client Guest Foreign Multicast error                    0                0
Wls Client FQDN filter error                                0                0
Wls Client IPSG v4 Ingress drop                             0                0
Wls Client IPSG v6 Invalid address drop                     1                94
Wls Client IPSG V6 entry already present error              0                0
Wls Client P2P blocking drop                                0                0
Wls Client iPSK P2P Tag Mismatch                            0                0
Wls Client Egress avc l2 fwd Error                          0                0
Wls Client Egress avc iv4 fwd Error                         0                0
Wls Client Egress avc iv6 fwd Error                         0                0
Wls Client block mgmt over wireless Error                   0                0
Wls Client block mgmt over wireless routed Error            0                0
Wls Client MDNS Packet Drop                                 0                0
```

Data Plane Health

Punt to Control Plane

# show platform hardware chassis active qfp feature wireless punt statistics

```
CPP Wireless Punt stats:
 App Tag                                          Packet Count
 -------                                          ------------
CAPWAP_PKT_TYPE_DOT11_PROBE_REQ      1253880
CAPWAP_PKT_TYPE_DOT11_MGMT           4
CAPWAP_PKT_TYPE_DOT11_IAPP                        792082
CAPWAP_PKT_TYPE_DOT11_RFID                        194627
CAPWAP_PKT_TYPE_DOT11_RRM            0
CAPWAP_PKT_TYPE_DOT11_DOT1X          0
CAPWAP_PKT_TYPE_CAPWAP_KEEPALIVE     246811
CAPWAP_PKT_TYPE_MOBILITY_KEEPALIVE   215591
CAPWAP_PKT_TYPE_CAPWAP_CNTRL         982084
CAPWAP_PKT_TYPE_CAPWAP_DATA          8
CAPWAP_PKT_TYPE_CAPWAP_DATA_PAT                   38
CAPWAP_PKT_TYPE_MOBILITY_CNTRL       68585
WLS_SMD_WEBAUTH                      0
SISF_PKT_TYPE_ARP                    45
SISF_PKT_TYPE_DHCP                   5
SISF_PKT_TYPE_DHCP6                  0
SISF_PKT_TYPE_IPV6_ND                12
SISF_PKT_TYPE_DATA_GLEAN                          0
SISF_PKT_TYPE_DATA_GLEAN_V6          0
SISF_PKT_TYPE_DHCP_RELAY                          5
WLCLIENT_PKT_TYPE_MDNS                            3012
CAPWAP_PKT_TYPE_CAPWAP_RESERVED                   0
```

Conclusion

Troubleshooting recap

Step 1 : Health Monitoring

```
# show wireless stats trace-on-failure
# show logging trace-on-failure summary / show logging profile wireless trace-on-failure
# show wireless stats ap join summary
# show wireless stats ap history
# show wireless stats client detail
```

Troubleshooting recap

Step 2 : Basic logging

# show log

```
Dec 18 13:38:18.228: %LINEPROTO-5-UPDOWN: Line protocol on Interface Capwap1, changed state to down
Dec 18 13:38:18.205: %CAPWAPAC_SMGR_TRACE_MESSAGE-3-EWLC_GEN_ERR: Chassis 1 R0/0: wncd: Error in Session-
IP: 192.168.16.134[5264] Mac: 7069.5a51.46e0 Heartbeat timer expiry for AP. Close CAPWAP DTLS session
Dec 18 13:38:18.231: %CAPWAPAC_SMGR_TRACE_MESSAGE-5-AP_JOIN_DISJOIN: Chassis 1 R0/0: wncd: AP Event: AP
Name: 4802paolo, MAC: 4c77.6d9e.60e4 Disjoined
Dec 21 06:19:45.425: %HTTP-4-SERVER_CONN_RATE_EXCEED: Number of connections per minute has exceeded the
maximum limit(500)as specified by the platform.
..Dec 21 06:20:00.748: %HTTP-4-SERVER_CONN_RATE_EXCEED: Number of connections per minute has exceeded the
maximum limit(500)as specified by the platform.
.Dec 21 06:20:00.785: %HTTP-4-SERVER_CONN_RATE_EXCEED: Number of connections per minute has exceeded the
maximum limit(500)as specified by the platform.
.Dec 21 06:20:15.616: %HTTP-4-SERVER_CONN_RATE_EXCEED: Number of connections per minute has exceeded the
maximum limit(500)as specified by the platform.
```

Troubleshooting recap

Step 3 : Pull always on data for a client/AP

```
# show logging profile wireless filter-mac <mac> to-file <filename> start last <minutes>
```

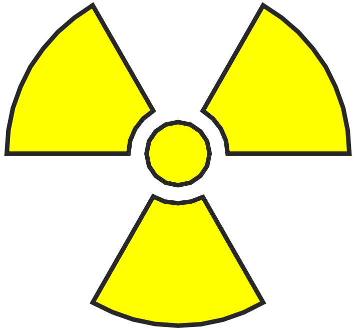
- Notice level data
- Logs will be rotated every 24/48h or more depending on platform and load

Troubleshooting recap

Step 4 : More information needed? RA Traces

```
#  debug wireless mac aaaa.bbbb.cccc monitor-time 10
```

Use the Web UI for it !



Troubleshooting recap

Step 5: Packet view needed? EPC

# monitor capture….FILENAME.pcap

Conclusion : troubleshooting recap

Step 5 : TAC case

- RA-trace output (internal level, while we're at it) or show logging profile wireless of always-on output filtered for the problematic mac or timestamp
- Relevant show techs (at least show tech + show tech wireless)
- Your observations from "show logging" or "show logging trace-on-failure summary" (timestamps, affected macs)
- Core dump files from the web UI troubleshooting page (if the problem is a crash)

Thank you