Troubleshoot Wireless LAN Controller CPU Load

Contents

Introduction

This document describes how to monitor CPU usage on Catalyst 9800 Wireless LAN Controllers, plus covers several configuration recommendations.

Understanding CPU Usage

Before you delve into CPU load troubleshooting, you need to understand the basics of how CPUs are used in Catalyst 9800 Wireless LAN Controllers, and some details on the software architecture.

In general, Catalyst 9800 Best Practices document defines a set of good configuration settings that can prevent application-level issues, for example, using location filtering for mDNS, or ensuring client exclusion is always enabled. It is advised that you apply those recommendations together with the topics exposed here.

Platform Basics

Catalyst 9800 controllers were designed as a flexible platform, targeting different network loads, and focusing on horizontal scaling. The internal development naming was “eWLC” with the e for “elastic”, to signify that the same software architecture, would be able to run from a small single CPU embedded system to multiple CPU/core large-scale appliances.

Each WLC has had two distinct “sides”:

- Control plane: handling all “management” interactions like CLI, UI, Netconf, and all onboarding processes for clients and APs.
- Data plane: responsible for actual packet forwarding, and decapsulation of CAPWAP, AVC policy enforcement, among other functionalities.

Control Plane

- Most Cisco IOS-XE processes run under BinOS (Linux Kernel), with its own specialized scheduler and monitoring commands.
- There is a set of key processes, called Wireless Network Control Daemon (WNCD) each having a local in-memory database, that handle most of the wireless activity. Each CPU owns a WNCD, to spread the load across all available CPU cores to each system
- Load distribution across WNCDs is done during AP join. When an AP performs a CAPWAP join to the controller, an internal load balancer distributes the AP using a set of possible rules, to ensure proper usage of all available CPU resources.
- Cisco IOS® code runs on its own process called IOSd, and has its CPU scheduler. This takes care of specific functionality, for example, CLI, SNMP, multicast and routing.

In a simplified view, the controller has communication mechanisms between the control and data plane, "punt”, sends traffic from the network to the control plane, and "injection", pushes frames from the control plane into the network.

As part of a possible high CPU troubleshooting investigation, you need to monitor the punt mechanism, to evaluate what traffic is reaching the control plane and could lead to a high load.

Data Plane

For the Catalyst 9800 controller, this is running as part of the Cisco Packet Processor (CPP), which is a software framework to develop packet forwarding engines, used across multiple products and technologies.

The architecture allows a common feature set, across different hardware or software implementations, for example, allowing similar features for 9800CL vs 9800-40, at different throughput scales.

AP Load Balancing

The WLC performs load balancing across CPUs during the CAPWAP AP join process, with the key differentiator being the AP site tag name. The idea is that each AP represents a specific CPU load added, coming from its client activity, and the AP itself. There are several mechanisms to perform this balancing:

- If the AP is using “default-tag”, it would be balanced in a round-robin fashion across all CPUs/WNCDs, with each new AP join going to the next WNCD. This is the simplest method, but it has few implications:
    - This is the suboptimal scenario, as APs in the same RF roaming domain, would do frequent Inter-WNCD roaming, involving additional inter-process communication. Roaming across instances is slower by a small percentage.
    - For the FlexConnect (remote) site tag, no PMK key distribution is available. This means that you can’t do fast roaming for Flex mode, impacting OKC/FT roaming modes.

      In general, the default tag can be used on lower load scenarios (for example less than 40% of AP and client load of the 9800 platform), and for FlexConenct deployment only when fast roaming is not a requirement.

- If the AP has a custom site tag, the first time an AP belonging to the site-tag name joins the controller, the site tag is assigned to a specific WNCD instance. All subsequent additional AP joins with the same tag are assigned to the same WNCD. This ensures roaming across APs in the same site-tag, happens in the one WCND context, which provides a more optimal flow, with lower CPU usage. Roaming across WNCDs is supported, just not as optimal as intra-WNCD roaming.

- Default load balancing decision: When a tag is assigned to a WNCD, the load balancer selects the instance with the lowest site tag count at that time. Because the total load that that site-tag might have is not known, it can lead to sub-optimal balancing scenarios. This depends on the order of AP joins, how many site-tags have been defined, and if the AP count is asymmetrical across them

- Static Load balancing: To prevent unbalanced site-tag assignment to WNCD, the site load command was introduced in 17.9.3 and higher, to allow administrators to predefine the expected load of each site tag. This is especially useful when handling campus scenarios, or multiple branch offices, each mapped to different AP counts, to ensure the load is evenly distributed across WNCD.

For example, if you have a 9800-40, handling one main office, plus 5 branch offices, with different AP counts, the configuration could look like this:

```
wireless tag site office-main
 load 120

wireless tag site branch-1
 load 10

wireless tag site branch-2
 load 12

wireless tag site branch-3
 load 45

wireless tag site branch-4
 load 80

wireless tag site branch-5
 load 5
```

In this scenario, you do not want the main office tag to be on the same WNCD as branch-3 and branch-4, there are in total 6 site tags, and the platform has 5 WNCDs, so there could be a chance that the highest loaded site tags land on the same CPU. By using the load command, you can create a predictable AP load balancing topology.

The load command is a expected size hint, it does not have to match exactly the AP count, but it is normally set to the expected APs that might join.

- In scenarios where you have large buildings handled by a single controller, it is easier and simpler to just create as many site-tag as WNCDs for that specific platform (for example., C9800-40 has five, C9800-80 has 8). Assign APs in the same area or roaming domain to the same site tags to minimize inter-WNCD communication.
- RF Load Balancing: This balances APs across WNCD instances, using the RF neighbor relationship from RRM, and creates sub-groups depending on how close the APs are to each other. It must be done after APs have been up and running for a while and remove the need to configure any static load balance settings. This is available from 17.12 and higher.

How to find out how many WNCDs are present?

For hardware platforms, the WNCD count is fixed: 9800-40 has 5, 9800-80 has 8. For 9800CL (virtual), the number of WNCDs would depend on the virtual machine template used during initial deployment.

As a general rule, if you want to find out how many WNCDs are running in the system, you can use this command across all controller types:

<#root>

```
9800-40#show processes cpu platform sorted  |  count wncd
Number of lines which match regexp =

5
```

In the case of 9800-CL specifically, you can use the command **show platform software system all** to collect details on the virtual platform:

<#root>

```
9800cl-1#show platform software system all
Controller Details:
=================
VM Template: small
Throughput Profile: low
AP Scale: 1000
Client Scale: 10000

WNCD instances: 1
```

Monitoring AP load balancing

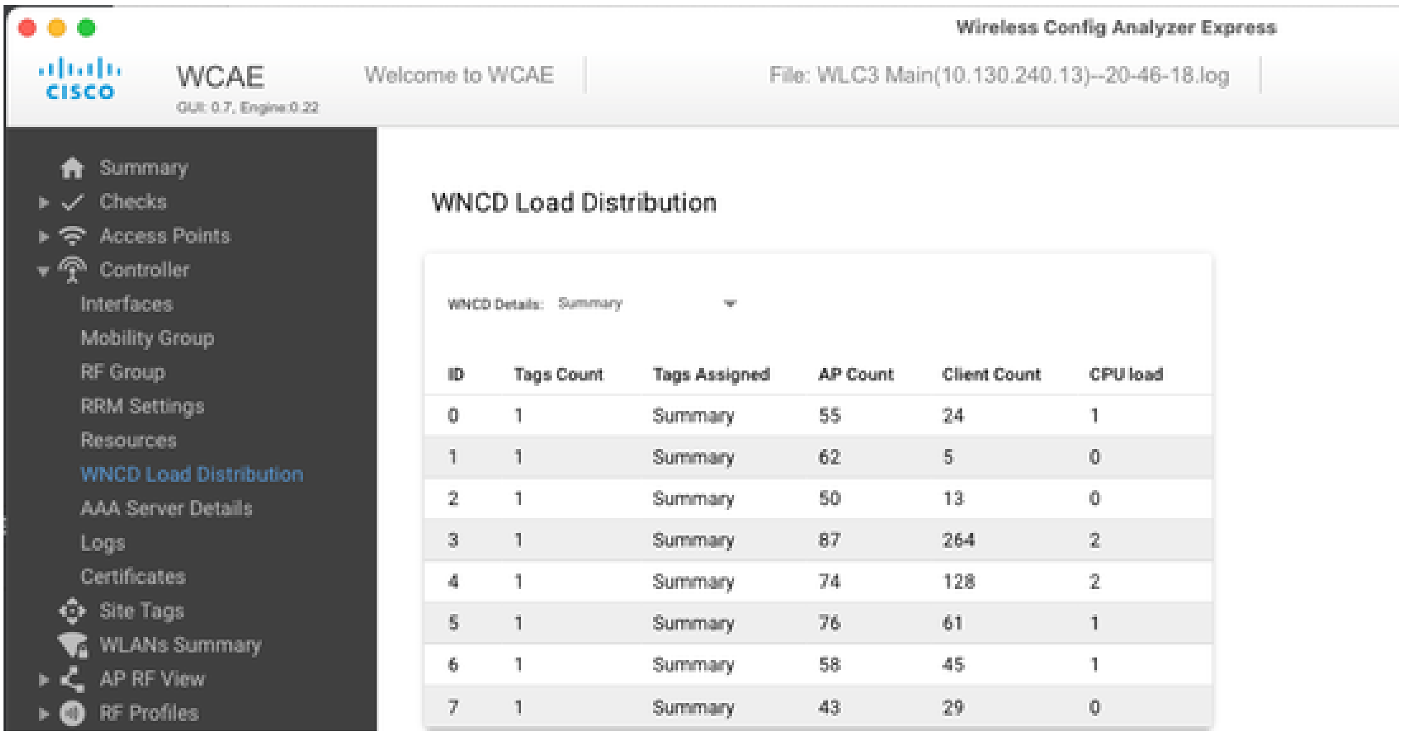
The AP to WNCD assignment is applied during the AP CAPWAP join process, so it is not expected to change during operations, regardless of the balancing method, unless there is a network-wide CAPWAP reset event where all APs disconnect and re-join.

The CLI command `show wireless loadbalance tag affinity` can provide an easy way to see the current state of AP load balance across all WNCD instances:

```
98001#show wireless loadbalance tag affinity
Tag                                 Tag type   No of AP's Joined    Load Config   Wncd Instance
-----------------------------------------------------------------------------------------------
Branch-tag                          SITE TAG   10                   0              0
Main-tag                            SITE TAG   200                  0              1
default-site-tag                    SITE TAG   1                    NA             2
```

if you want to correlate the AP distribution, against client count and CPU load, the easiest way is to use the WCAE support tool and load a `show tech wireless` taken during busy times. The tool summarizes the WNCD client count, taken from each AP that is associated with it.

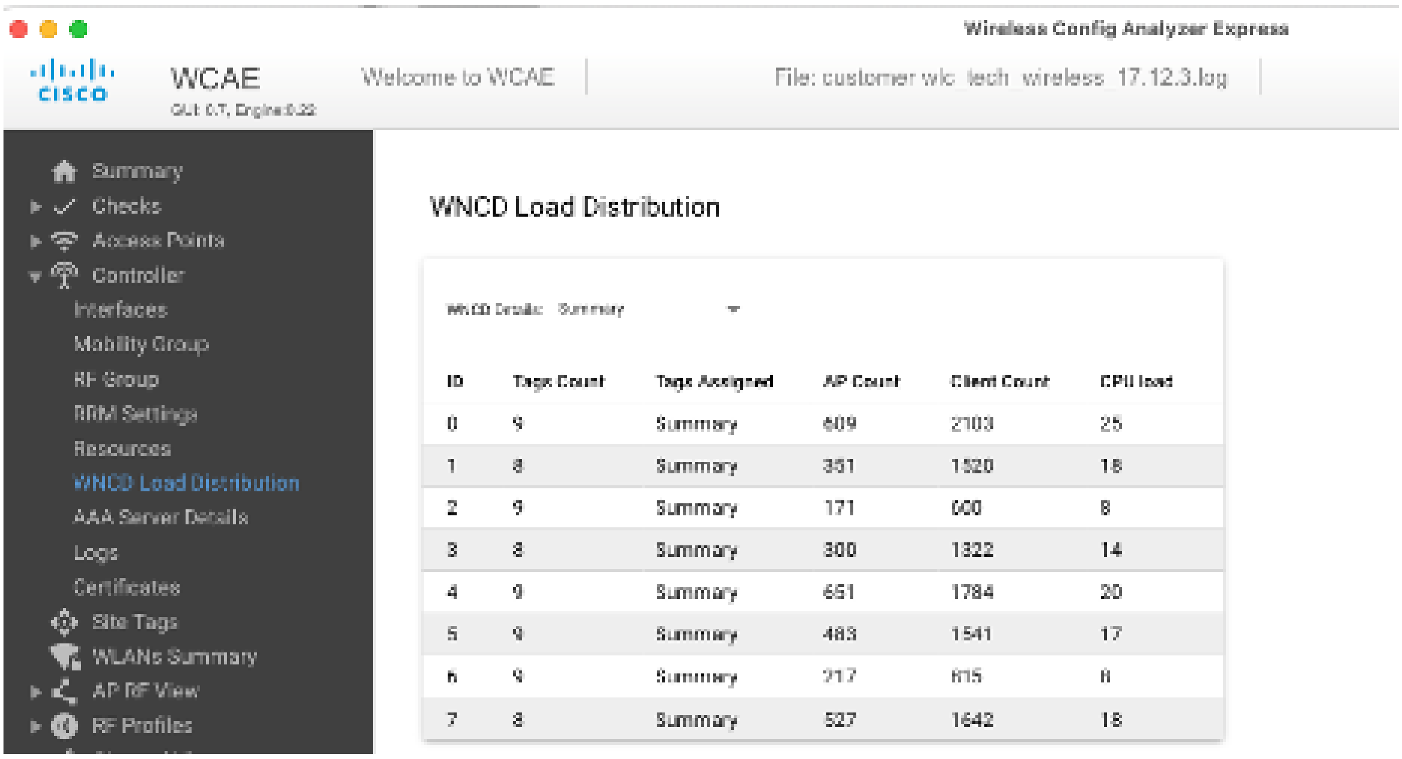
Example of a properly balanced controller, during low usage and client count:



WNCD Load Distribution

ID	Tags Count	Tags Assigned	AP Count	Client Count	CPU load
0	1	Summary	55	24	1
1	1	Summary	62	5	0
2	1	Summary	50	13	0
3	1	Summary	87	264	2
4	1	Summary	74	128	2
5	1	Summary	76	61	1
6	1	Summary	58	45	1
7	1	Summary	43	29	0

Another example, for a more loaded controller, showing normal CPU utilization:



WNCD Load Distribution

ID	Tags Count	Tags Assigned	AP Count	Client Count	CPU load
0	9	Summary	609	2103	25
1	8	Summary	351	1520	18
2	9	Summary	171	600	8
3	8	Summary	300	1322	14
4	9	Summary	651	1784	20
5	9	Summary	483	1541	17
6	9	Summary	217	615	8
7	8	Summary	527	1642	18

What is the recommended AP load-balancing mechanism?

In short, you can summarize the different options in:

- Small network, no need for fast roaming, less than 40% of controller load: Default tag.
- If fast-roaming is needed (OKC, FT, CCKM), or large client count:
  - Single building: create as many site tags as CPUs (platform dependent)

- Before 17.12, or less than 500 AP count: Multiple buildings, branches or large campus: Create a site-tag per physical RF location, and configure load command per site.
- 17.12 and higher with more than 500 APs: use RF load balancing.

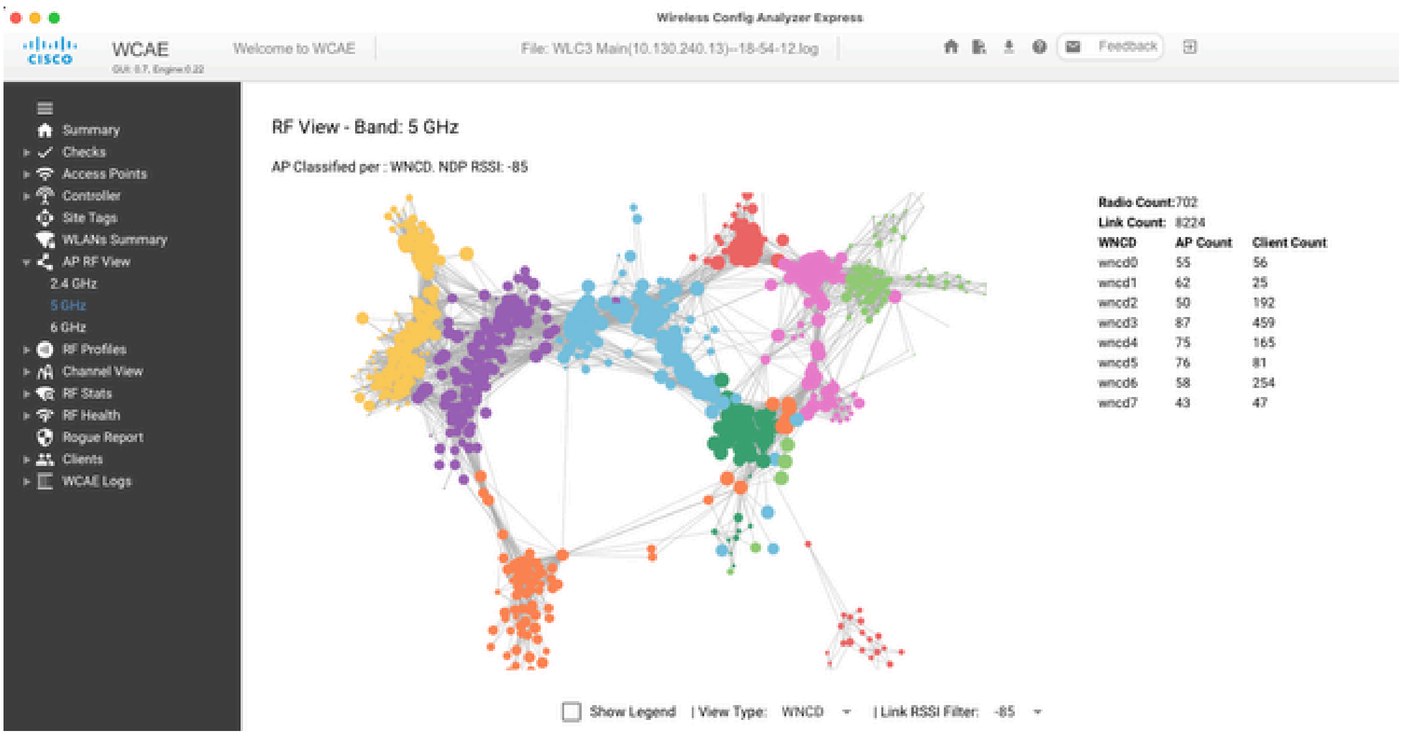
This 500 AP threshold, s to mark when it is effective to apply the load balancing mechanism, as it groups APs in blocks of 100 units by default.

AP WNCD Distribution Visualization

There are scenarios where you want to do a more advanced AP balancing, and it is desirable to have granular control on how APs are spread across CPUs, for example, very high-density scenarios where the key load metric is client count versus just purely focusing on the number of APs present in the system.

A good example of this situation is large events: a building could be hosting thousands of clients, over several hundred APs, and you would need to split the load across as many CPUs as possible but optimise roaming at the same time. So, you don't roam across WNCD unless it is needed. You want to prevent "salt & pepper" situations where multiple APs in different WNCDs/site tags are intermixed in the same physical location.

To help fine-tune, and provide a visualization of the distribution, you can use the WCAE tool, and take advantage of the AP RF View feature:



RF View - Band: 5 GHz

AP Classified per : WNCD. NDP RSSI: -85

Radio Count: 702
Link Count: 8224

WNCD	AP Count	Client Count
wncd0	55	56
wncd1	62	25
wncd2	50	192
wncd3	87	459
wncd4	75	165
wncd5	76	81
wncd6	58	254
wncd7	43	47

This allows us to see AP/WNCD distribution, just set `View Type` to WNCD. Here each color would represent a WNCD/CPU. You can also set the RSSI filter to -85, to avoid low signal connections, that are also filtered by the RRM algorithm in the controller.

In the previous example, corresponding to Ciscolive EMEA 24, you can see that most adjacent APs are clustered nicely across in the same WNCD, with very limited cross-overlapping.

Site tags allocated to the same WNCD, get the same color.

Monitoring Control Plane CPU Usage

It is important to remember the concept of Cisco IOS-XE architecture and keep in mind that there are two main “views” of CPU usage. One comes from historical Cisco IOS support, and the main one, with a holistic view of the CPU across all processes and cores.

In general, you can use the command `show processes cpu platform sorted` to collect detailed information for all processes across Cisco IOS-XE:

```
9800cl-1#show processes cpu platform sorted

CPU utilization for five seconds:  8%, one minute: 14%, five minutes: 11%
Core 0: CPU utilization for five seconds:  6%, one minute: 11%, five minutes:  5%
Core 1: CPU utilization for five seconds:  2%, one minute:  8%, five minutes:  5%
Core 2: CPU utilization for five seconds:  4%, one minute: 12%, five minutes: 12%
Core 3: CPU utilization for five seconds: 19%, one minute: 23%, five minutes: 24%
   Pid    PPid    5Sec    1Min    5Min  Status        Size  Name
--------------------------------------------------------------------------------
 19953   19514     44%     44%     44%  S           190880  ucode_pkt_PPE0
 28947    8857      3%     10%      4%  S          1268696  linux_iosd-imag
 19503   19034      3%      3%      3%  S           247332  fman_fp_image
 30839       2      0%      0%      0%  I                0  kworker/0:0
 30330   30319      0%      0%      0%  S             5660  nginx
 30329   30319      0%      1%      0%  S            20136  nginx
 30319   30224      0%      0%      0%  S            12480  nginx
 30263       1      0%      0%      0%  S             4024  rotee
 30224    8413      0%      0%      0%  S             4600  pman
 30106       2      0%      0%      0%  I                0  kworker/u11:0
 30002       2      0%      0%      0%  S                0  SarIosdMond
 29918   29917      0%      0%      0%  S             1648  inet_gethost
```

There are several key points to highlight here:

- The process ucode_pkt_PPE0 is handling the data plane on 9800L and 9800CL platforms, and it is expected to see a high utilization all the time, even higher than 100%. This is part of implementation, and this does not constitute a problem.
- It is important to differentiate peak usage vs a sustained load and isolate what is expected in a given scenario. For example, collecting a very large CLI output, like `show tech wireless` can generate a peak load on IOSd, smand, pubd processes, as a very large text output is being collected, with hundreds of CLI commands executed, this is not a problem, and the load goes down after the output has been completed.

```
   Pid    PPid    5Sec    1Min    5Min  Status        Size  Name
--------------------------------------------------------------------------------
 19371   19355     62%     83%     20%  R           128120  smand
 27624   27617     53%     59%     59%  S          1120656  pubd
  4192    4123     11%      5%      4%  S          1485604  linux_iosd-imag
```

- Peak utilization for WNCD cores is expected, during high client activity times. It is possible to see peaks of 80%, without any functional impact, and they normally do not constitute a problem.

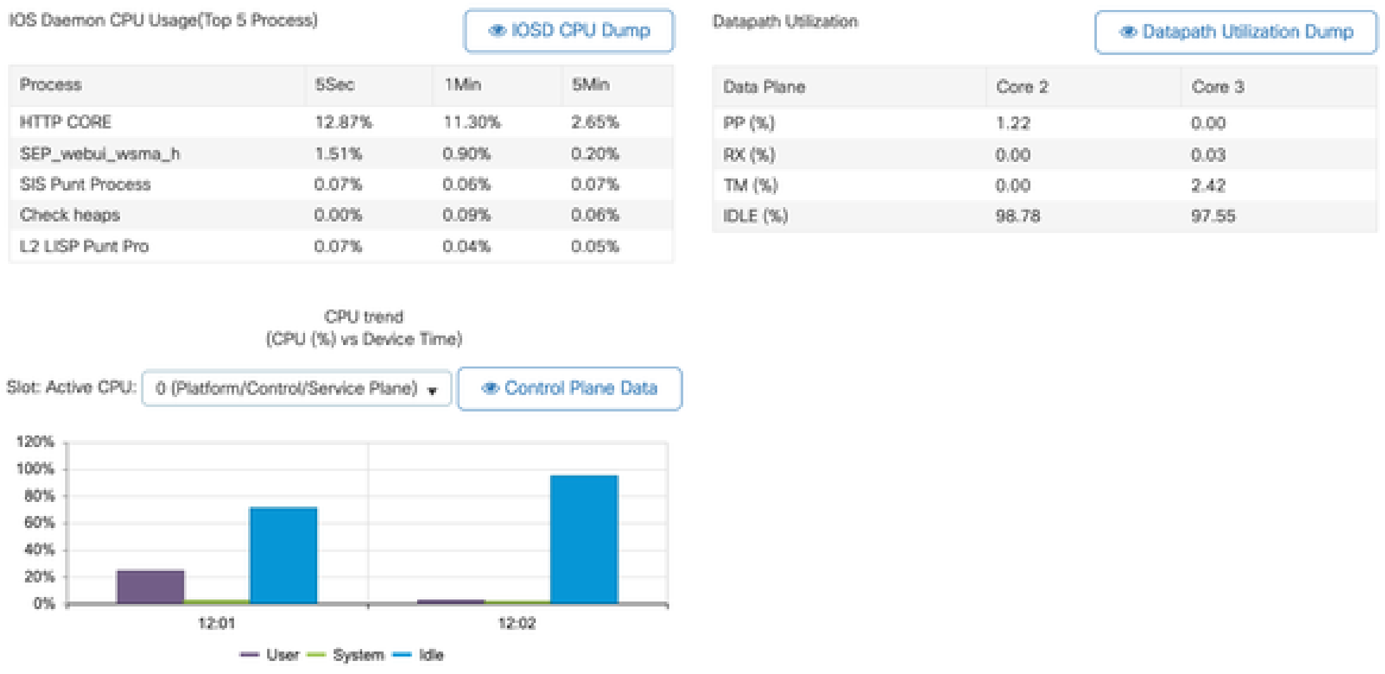
```
   Pid    PPid    5Sec    1Min    5Min  Status        Size  Name
--------------------------------------------------------------------------------
 21094   21086     25%     25%     25%  S           978116  wncd_0
 21757   21743     21%     20%     20%  R          1146384  wncd_4
 22480   22465     18%     18%     18%  S          1152496  wncd_7
 22015   21998     18%     17%     17%  S           840720  wncd_5
 21209   21201     16%     18%     18%  S           779292  wncd_1
 21528   21520     14%     15%     14%  S           926528  wncd_3
```

- A sustained high CPU usage on a process, higher than 90%, for more than 15 minutes, must be investigated.
- You can monitor IOSd CPU utilization, with the command `show processes cpu sorted` . This corresponds to activity in the linux_iosd-imag process part of the Cisco IOS-XE list.

```
9800cl-1#show processes cpu sorted

CPU utilization for five seconds: 2%/0%; one minute: 3%; five minutes: 3%
 PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
 215          81          88        920  1.51%  0.12%  0.02%   1 SSH Process
 673      164441     7262624         22  0.07%  0.00%  0.00%   0 SBC main process
 137     2264141   225095413         10  0.07%  0.04%  0.05%   0 L2 LISP Punt Pro
 133      534184    21515771         24  0.07%  0.04%  0.04%   0 IOSXE-RP Punt Se
 474     1184139    56733445         20  0.07%  0.03%  0.00%   0 MMA DB TIMER
   5           0           1          0  0.00%  0.00%  0.00%   0 CTS SGACL db cor
   6           0           1          0  0.00%  0.00%  0.00%   0 Retransmission o
   2      198433      726367        273  0.00%  0.00%  0.00%   0 Load Meter
   7           0           1          0  0.00%  0.00%  0.00%   0 IPC ISSU Dispatc
  10     3254791      586076       5553  0.00%  0.11%  0.07%   0 Check heaps
   4          57          15       3800  0.00%  0.00%  0.00%   0 RF Slave Main Th
   8           0           1          0  0.00%  0.00%  0.00%   0 EDDRI_MAIN
```

- You can use the 9800 GUI, for a quick view of IOSd load, per core usage, and data plane load:



Process	5Sec	1Min	5Min
HTTP CORE	12.87%	11.30%	2.65%
SEP_webui_wsma_h	1.51%	0.90%	0.20%
SIS Punt Process	0.07%	0.06%	0.07%
Check heaps	0.00%	0.09%	0.06%
L2 LISP Punt Pro	0.07%	0.04%	0.05%

Data Plane	Core 2	Core 3
PP (%)	1.22	0.00
RX (%)	0.00	0.03
TM (%)	0.00	2.42
IDLE (%)	98.78	97.55

This is available on the Monitoring/System/CPU Utilization tab.

What is each process?

The exact process list would vary depending on the controller model, and the Cisco IOS-XE version. This is a list of some of the key processes, and it is not intended to cover all possible entries.

Process Name	**What does it do?**	**Evaluation**
wncd_x	Handles most wireless operations. Depending on the 9800 model, you can have between 1 to 8 instances	You could see peaks of high utilization during busy hours. Report if the utilization is stuck a 95% or more for several minutes
linux_iosd-imag	IOS process	Expected to see high utilization if collecting large CLI output (show tech) Large or too frequent SNMP operations can lead to high CPU
nginx	Web server	This process can show peaks, and should only be reported on a sustained high load
ucode_pkt_PPE0	Data plane in 9800CL/9800L	Use the command **show platform hardware chassis active qfp datapath utilization** to monitor this component

ezman	Chipset manager for interfaces	A sustained high CPU here could indicate either an HW issue or a possible kernel software problem. It should be reported
dbm	Database Manager	A sustained high CPU here should be reported
odm_X	Operation Data Manager handles consolidated DB across processes	High CPU expected on loaded systems
rogued	Handles Rogue functionality	A sustained high CPU here should be reported
smand	Shell Manager. Takes care of CLI parsing, and interaction across different processes	High CPU expected while handling large CLI output. Sustained high CPU in the absence of load should be reported
emd	Shell Manager. Takes care of CLI parsing, and interaction across different processes	High CPU expected while handling large CLI output. Sustained high CPU on the absence of load should be reported
pubd	Part of telemetry handling	High CPU expected for large telemetry subscriptions. Sustained high CPU on the absence of load should be reported

High CPU Protection Mechanisms

Catalyst 9800 Wireless LAN Controllers have extensive protection mechanisms around network or wireless client activity, to prevent high CPU due to accidental or intentional scenarios. There are several key features designed to help you contain problematic devices:

Client Exclusion

This is enabled by default, and is part of Wireless Protection Policies, and it can be enabled or disabled per Policy Profile. This can detect several different behavior issues, remove the client from the network, and set it into a “temporary exclusion list”. While the client is on this excluded state, the APs do not talk to them, preventing any further actions.

After the exclusion timer has passed (60 seconds by default), the client is allowed to associate again.

There are several triggers for client exclusion:

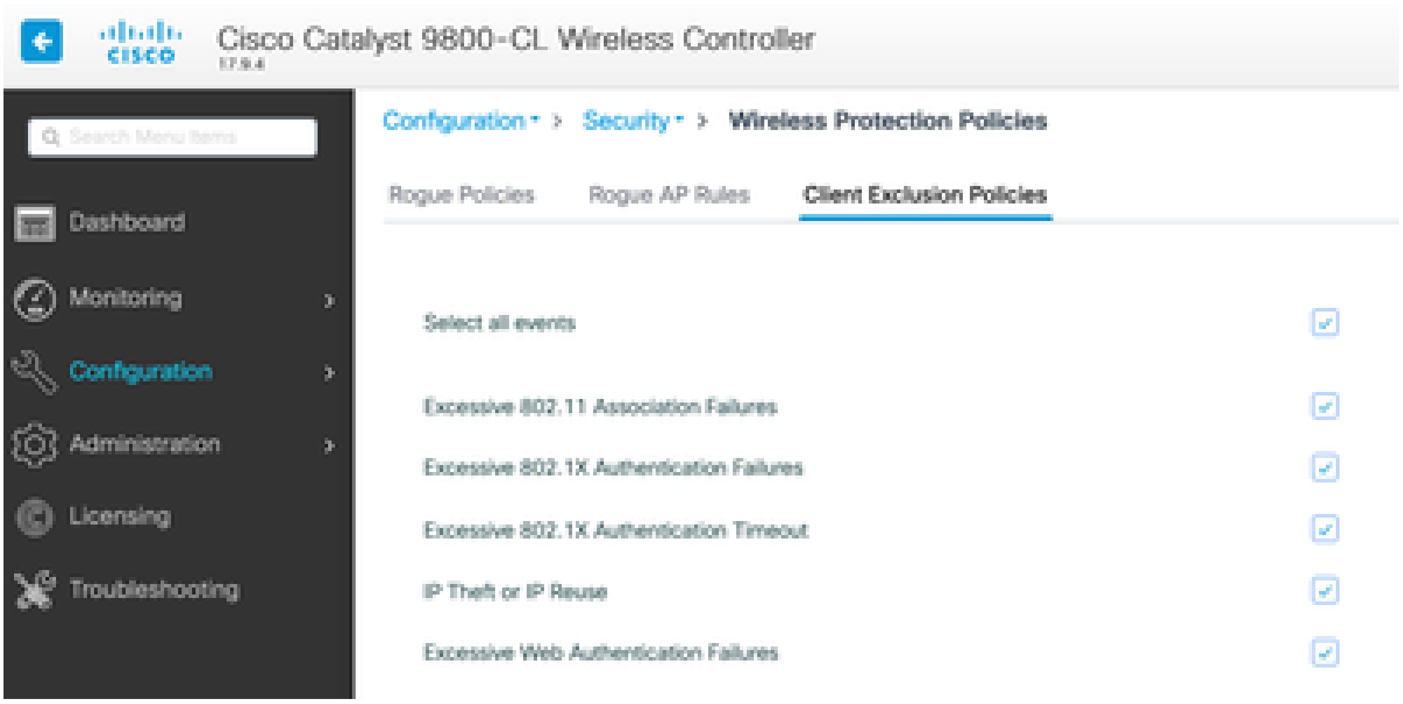
- Repeated association failures

- 3 or more webauth, PSK or 802.1x authentication errors
- Repeated authentication timeouts (no response from client)
- Trying to reuse an IP address, already registered to another client
- Generating an ARP flood

Client exclusion protects your controller, AP and AAA infrastructure(Radius) from several high activity types that could lead to a high CPU. In general, it is not advisable to disable any of the exclusion methods, unless needed for a troubleshooting exercise or compatibility requirement.

The default settings work for almost all cases, and only on some exceptional scenarios, is needed to increase the exclusion time, or disable some specific trigger. For example, some legacy or specialized clients (IOT/Medical), might need to have the association failure trigger to be disabled, due to client-side defects that can not be easily patched

You can customize the triggers in the UI: Configuration/Wireless Protection/Client Exclusion Policies:


Cisco Catalyst 9800-CL Wireless Controller
17.9.4
Search Menu Items
Dashboard
Monitoring
Configuration
Administration
Licensing
Troubleshooting
Configuration > Security > Wireless Protection Policies
Rogue Policies
Rogue AP Rules
Client Exclusion Policies

Select all events	✓
Excessive 802.11 Association Failures	✓
Excessive 802.1X Authentication Failures	✓
Excessive 802.1X Authentication Timeout	✓
IP Theft or IP Reuse	✓
Excessive Web Authentication Failures	✓

ARP Exclusion trigger was designed to be permanently enabled at a global level, but it can be customized on each policy profile. You can check the status with the command `sh wireless profile policy all` look for this specific output:

```
ARP Activity Limit
  Exclusion                            : ENABLED
  PPS                                  : 100
  Burst Interval                       : 5
```

Control Plane protection from Data Traffic

This is an advanced mechanism in the Data Plane, to ensure that traffic sent to Control Plane does not exceed a predefined set of thresholds. The feature is called “Punt Policers” and in almost all scenarios, it is not necessary to touch them, and even then, only must be done while working together with Cisco Support.

The advantage of this protection is that it provides a very detailed insight into what is going on in the network, and if there is any specific activity that is having an increased rate, or unexpectedly high packets per second.

This is only exposed through CLI, as they are normally part of advanced functionality that is rarely needed to modify.

To get a view of all punt policies:

```
9800-l#show platform software punt-policer

Per Punt-Cause Policer Configuration and Packet Counters
Punt                                Config Rate(pps)     Conform Packets                   Dropped Pack
Cause  Description                   Normal   High     Normal           High             Normal
-------------------------------------------------------------------------------------------------------
  2    IPv4 Options                  874      655      0                0                0
  3    Layer2 control and legacy     8738     2185     33               0                0
  4    PPP Control                   437      1000     0                0                0
  5    CLNS IS-IS Control            8738     2185     0                0                0
  6    HDLC keepalives               437      1000     0                0                0
  7    ARP request or response       437      1000     0                330176           0
  8    Reverse ARP request or repso  437      1000     0                24               0
  9    Frame-relay LMI Control       437      1000     0                0                0
 10    Incomplete adjacency          437      1000     0                0                0
 11    For-us data                   40000    5000     442919246        203771           0
 12    Mcast Directly Connected Sou  437      1000     0                0                0
```

This might be a large list, with more than 160 entries, depending on the software version.

On the table output, you want to check the dropped packet column along with any entry that has a non-zero value on the high drop count.

To simplify the data collection, you can use the command `show platform software punt-policer drop-only` , to filter for only policer entries with drops.

This feature could be useful to identify if there are ARP storms or 802.11 probe floods (they use queue “802.11 Packets to LFTS”. LFTS stands for Linux Forwarding Transport Service).

Wireless Call Admission Control

In all recent maintenance releases, the controller has an activity monitor, to dynamically react to high CPU, and ensure AP CAPWAP tunnels remain active, in the face of unsustainable pressure.

The feature checks on the WNCD load, and starts throttling new client activity, to ensure enough resources remain to handle the existing connections and protect CAPWAP stability.

This is enabled by default, and it does not have configuration options.

There are three levels of protection defined, L1 at 80% load, L2 at 85% load, and L3, at 89%, each one triggering different incoming protocol drops as protection mechanisms. The protection is automatically removed, as soon as the load decreases.

In a healthy network, you should not see L2 or L3 load events, and if they are happening frequently, it should be investigated.

To monitor use the command `wireless stats cac` as shown in the image.

```
9800-l#  show wireless stats cac

WIRESLESS CAC STATISTICS
---------------------------------------------
L1 CPU Threshold: 80         L2 CPU Threshold: 85         L3 CPU Threshold: 89
Total Number of CAC throttle due to IP Learn:  0
Total Number of CAC throttle due to AAA:      0
Total Number of CAC throttle due to Mobility Discovery:  0
Total Number of CAC throttle due to IPC:      0
CPU Throttle Stats
      L1-Assoc-Drop:    0          L2-Assoc-Drop:    0          L3-Assoc-Drop: 0
      L1-Reassoc-Drop:  0          L2-Reassoc-Drop:  0          L3-Reassoc-Drop: 0
      L1-Probe-Drop:    12231      L2-Probe-Drop:    11608      L3-Probe-Drop: 93240
      L1-RFID-Drop:     0          L2-RFID-Drop:     0          L3-RFID-Drop: 0
      L1-MDNS-Drop:     0          L2-MDNS-Drop:     0          L3-MDNS-Drop: 0
```

mDNS Protections

mDNS as a protocol allows a “zero-touch” approach to discover services across devices, but at the same time, it can be very active, and drive load significantly, if not configured properly.

mDNS, without any filtering, can easily drive up WNCD CPU utilization, coming from several factors:

1. mDNS policies with unrestricted learning, the controller will obtain all services offered by all devices. This can lead to very large service lists, with hundreds of entries.
2. Policies set without filtering: this will cause the controller to push those large service lists, to each client that asks who is providing a given service.
3. Some mDNS-specific services are provided by “all” wireless clients, leading to higher service count and activity, with variations on this by OS version.

You can check mDNS list size per service with this command:

```
9800-l#  show mdns-sd service statistics
Service Name                                                Service Count
-----------------------------------------------------------------------------
_ipp._tcp.local                                             84
_ipps._tcp.local                                            52
_raop._tcp.local                                            950
_airplay._tcp.local                                         988
_printer._tcp.local                                         13
_googlerpc._tcp.local                                       12
_googlecast._tcp.local                                      70
_googlezone._tcp.local                                      37
_home-sharing._tcp.local                                    7
_cups._sub._ipp._tcp.local                                  26
```

This can provide an idea of how large can get any given query, it does not denote a problem by itself, just a way to monitor what is tracked.

There are some important mDNS configuration recommendations:

- Set mDNS transport to a single protocol:

```
9800-1(config)# mdns-sd gateway

9800-1(config-mdns-sd)# transport ipv4
```

By default, it uses IPv4 transport, for performance, it is advisable to use either IPv6 or IPv4, but not both:

- Always set a location filter in the mDNS service policy, to avoid unbound queries/responses. In general, it is recommended to use “site-tag” but other options could work, depending on your needs.

I need more help

In case you are seeing high CPU load, and none of the above helps, please contact CX through a case, and add this data as starting point:

- Base data, as this would include AP/Controller config, and network and RF operational values:

```
show tech-support wireless
```

- Archive for all controller traces. This is a large file, simiilar to a "blackbox" concept, that can be collected with the command:

```
request platform software trace archive last <days> to-file bootflash:<archive file>
```