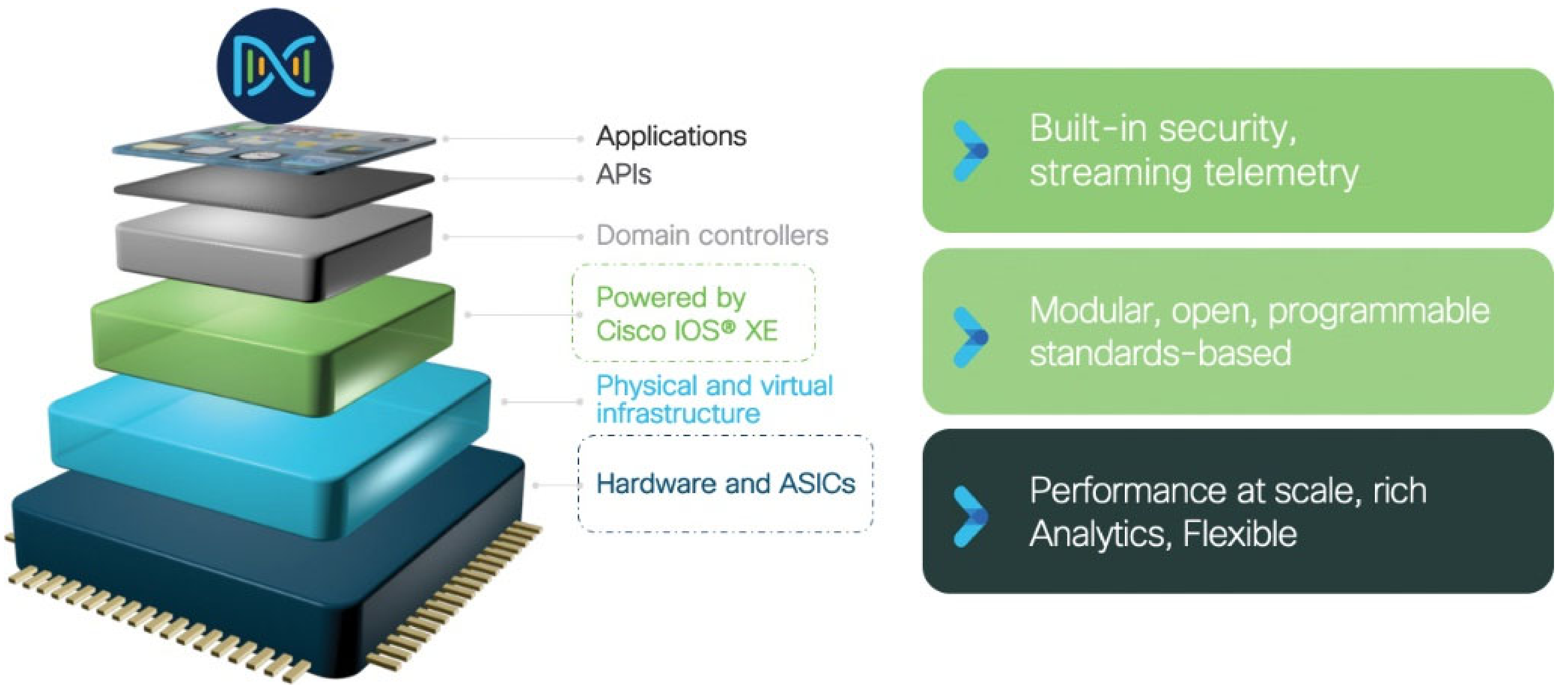


Figure 1-1 The principles of intent-based networking

From *Understanding and Troubleshooting Cisco Catalyst 9800 Series Wireless Controller by* Simone Arena, Francisco Sedano Crippa, Nicolas Darchis and Sudha Katgeri (9780137492329) Cisco Press

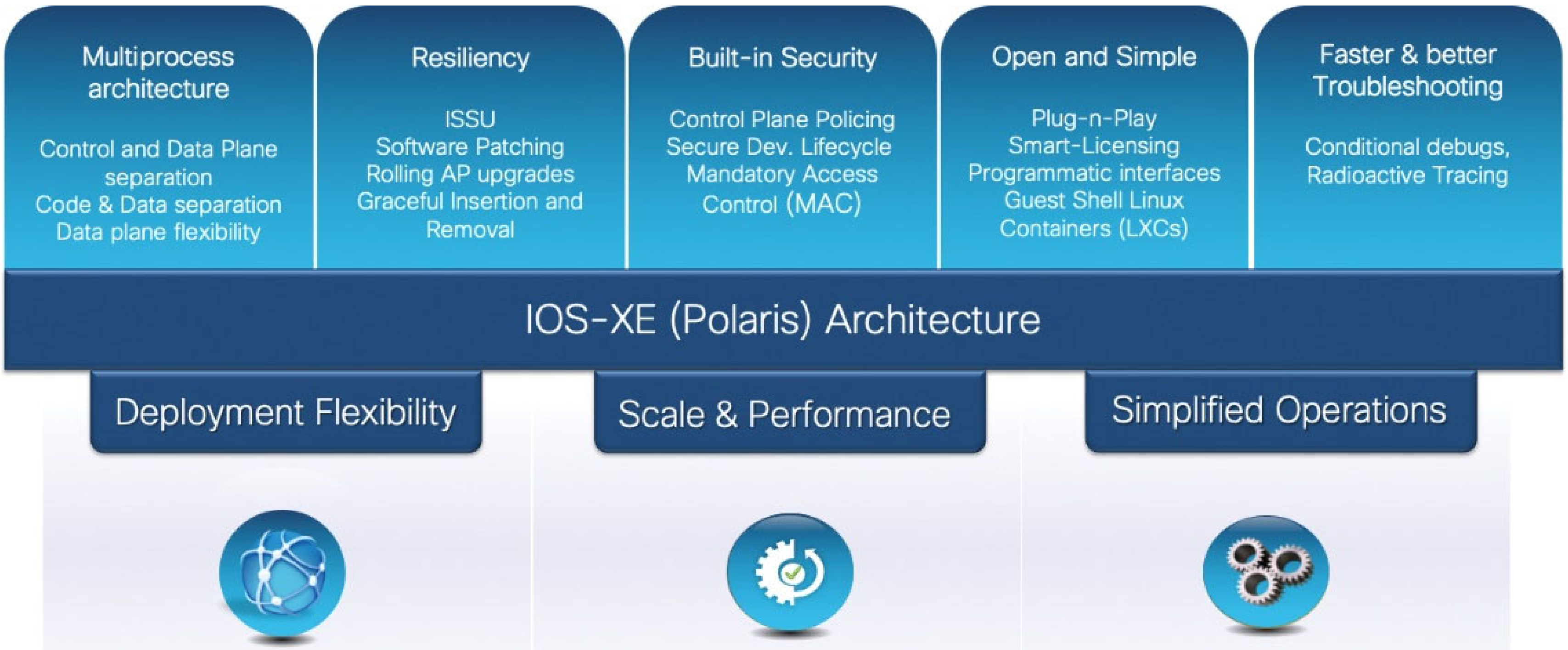


Figure 1-2 IOS-XE: A modern networking operating system

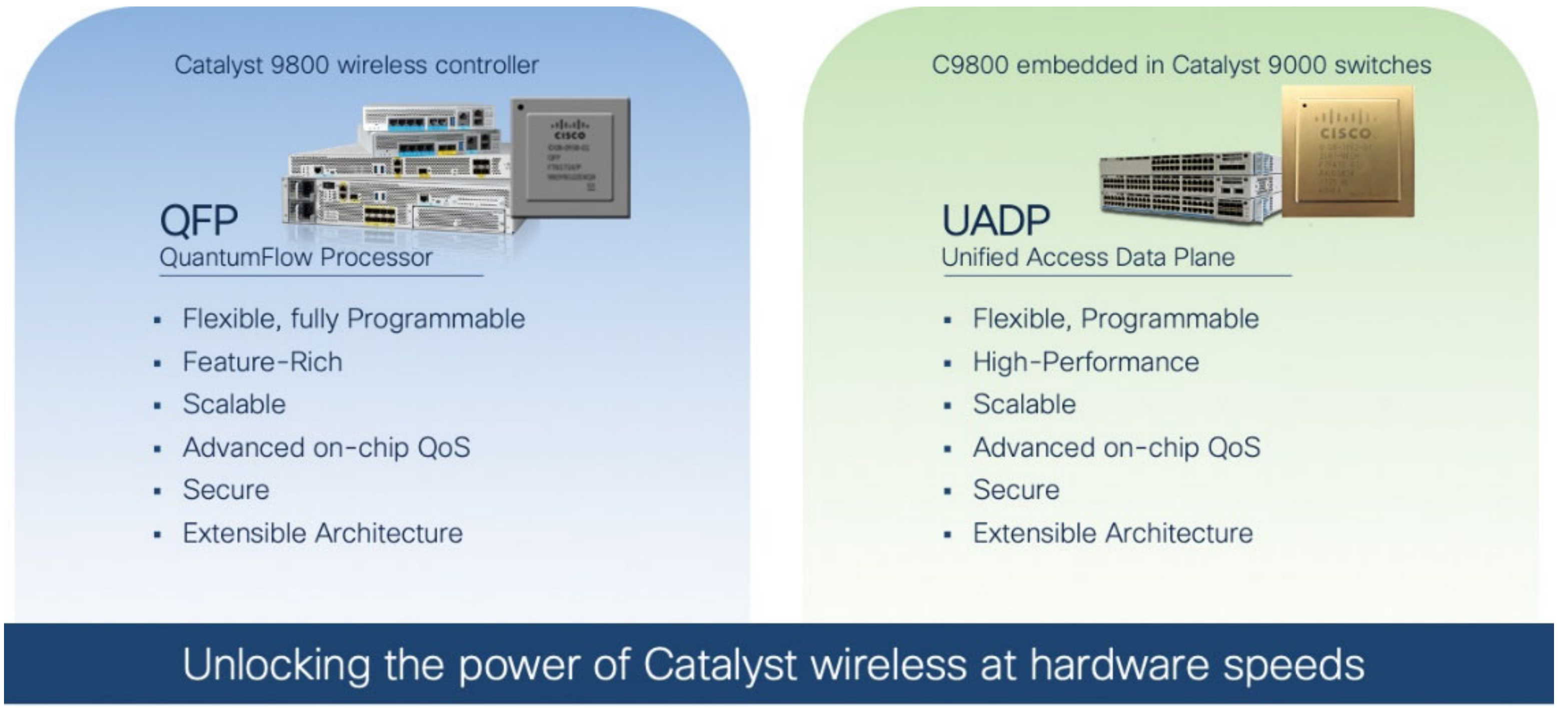


Figure 1-3 Catalyst 9800 is built on programmable silicon

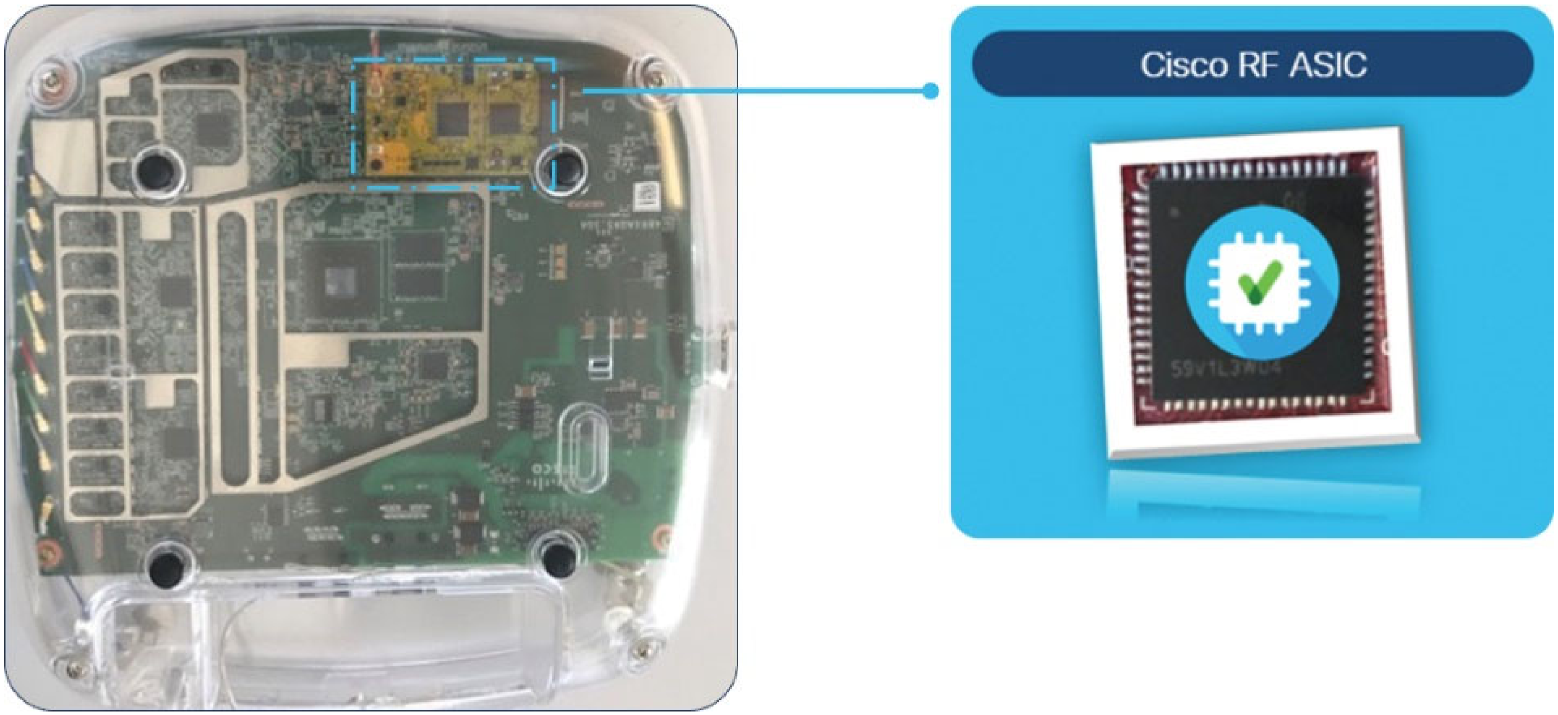


Figure 1-4 Cisco RF ASIC: Software-defined radio using a Mini-PCIe slot

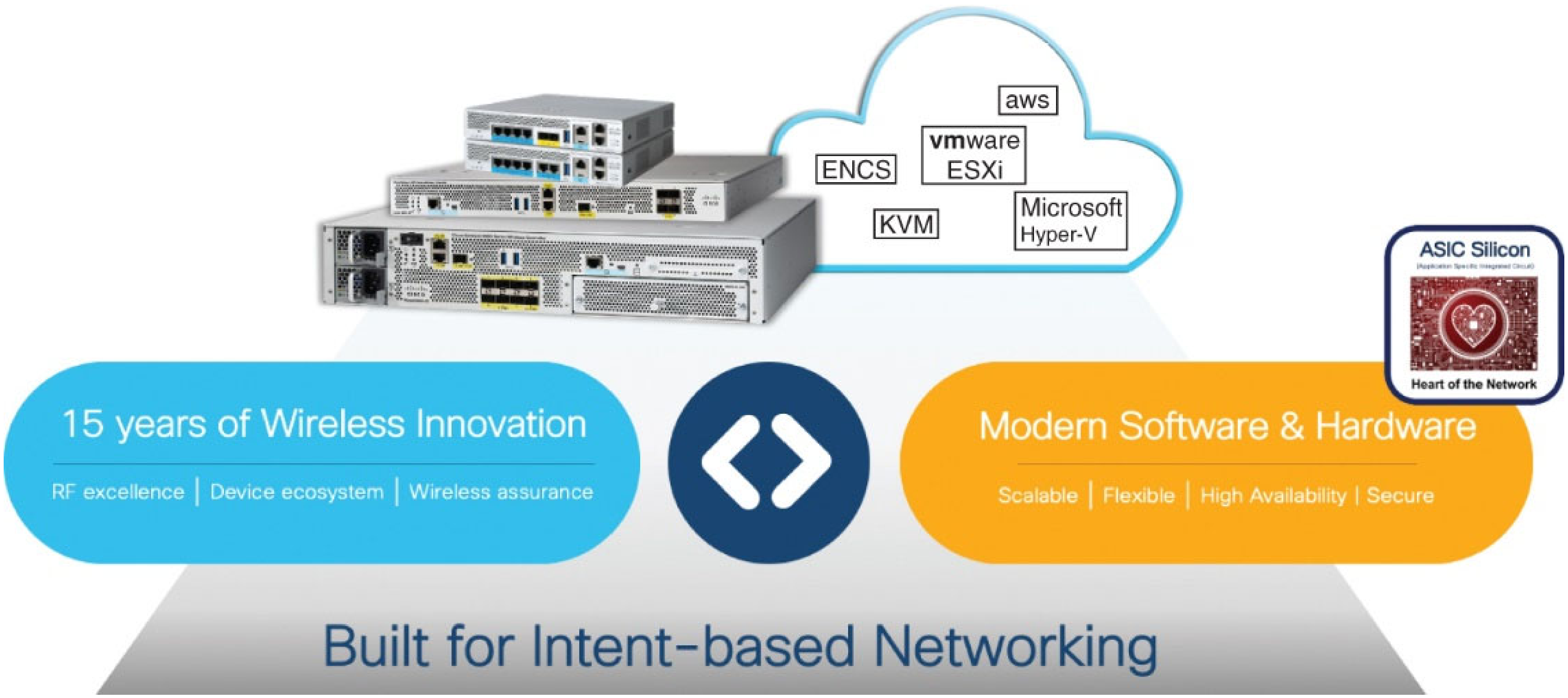


Figure 1-5 Catalyst 9800 built for intent-based networking

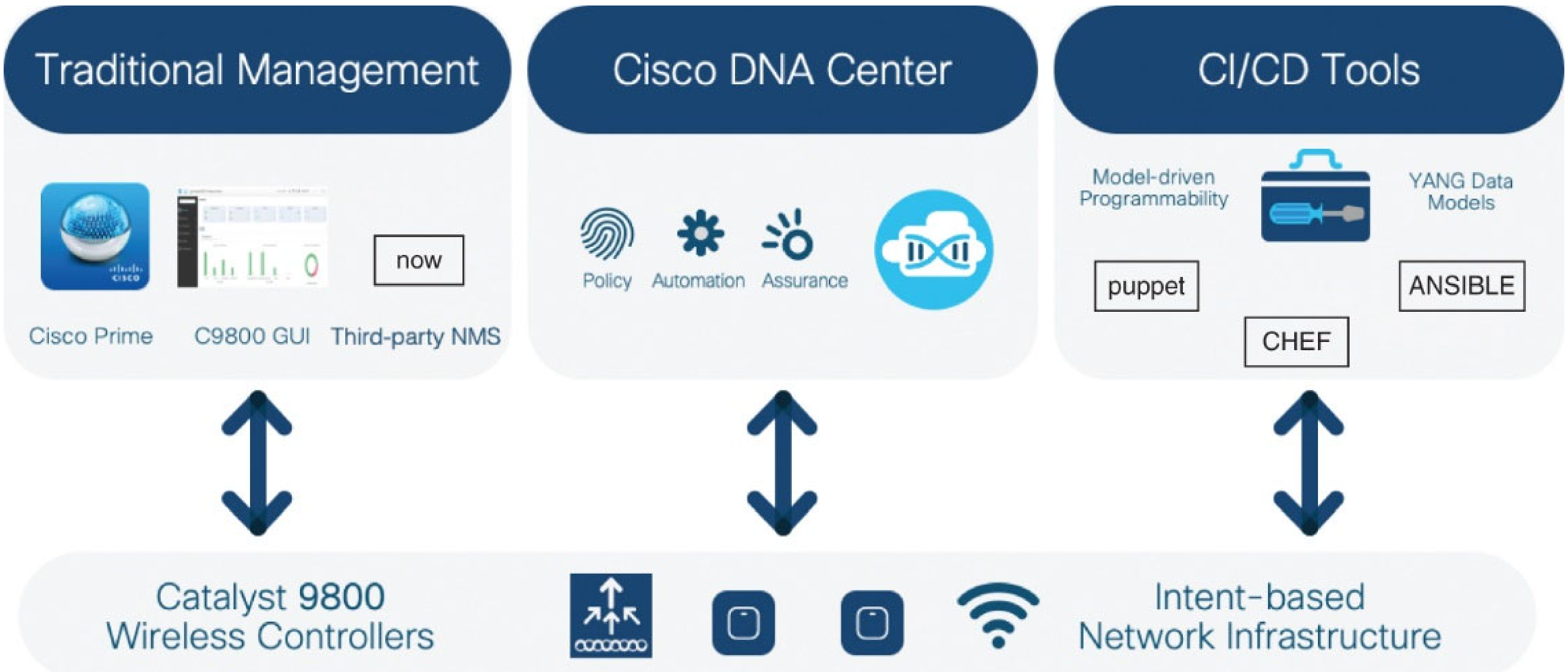


Figure 1-6 Catalyst 9800 flexible management options

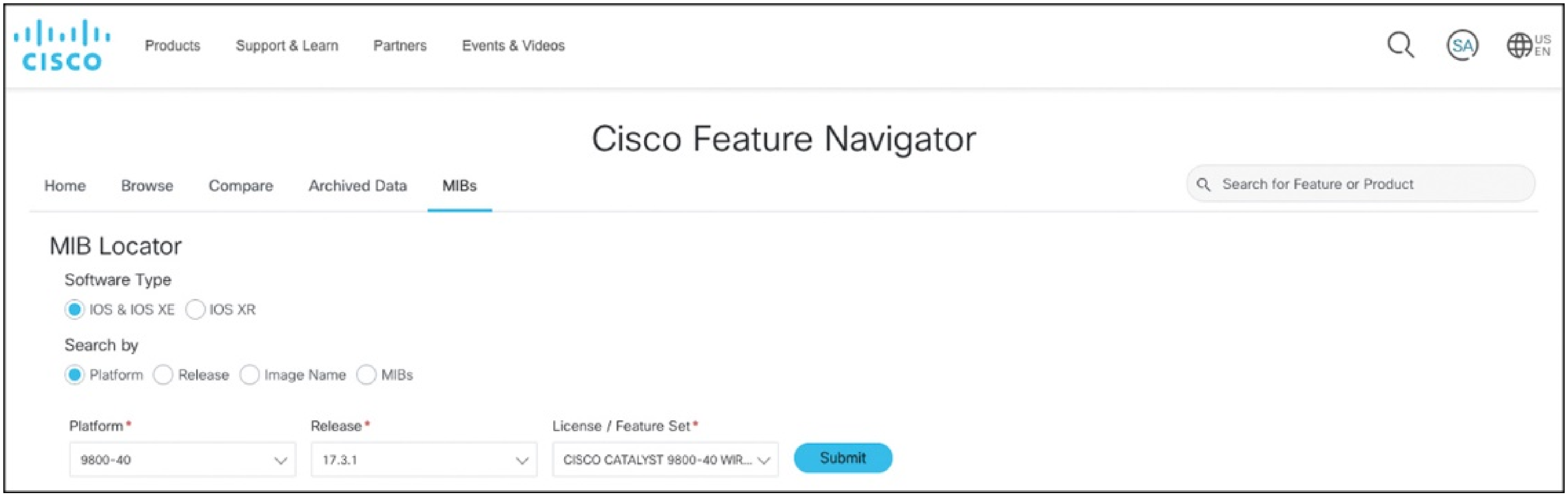


Figure 1-7 Cisco Feature Navigator

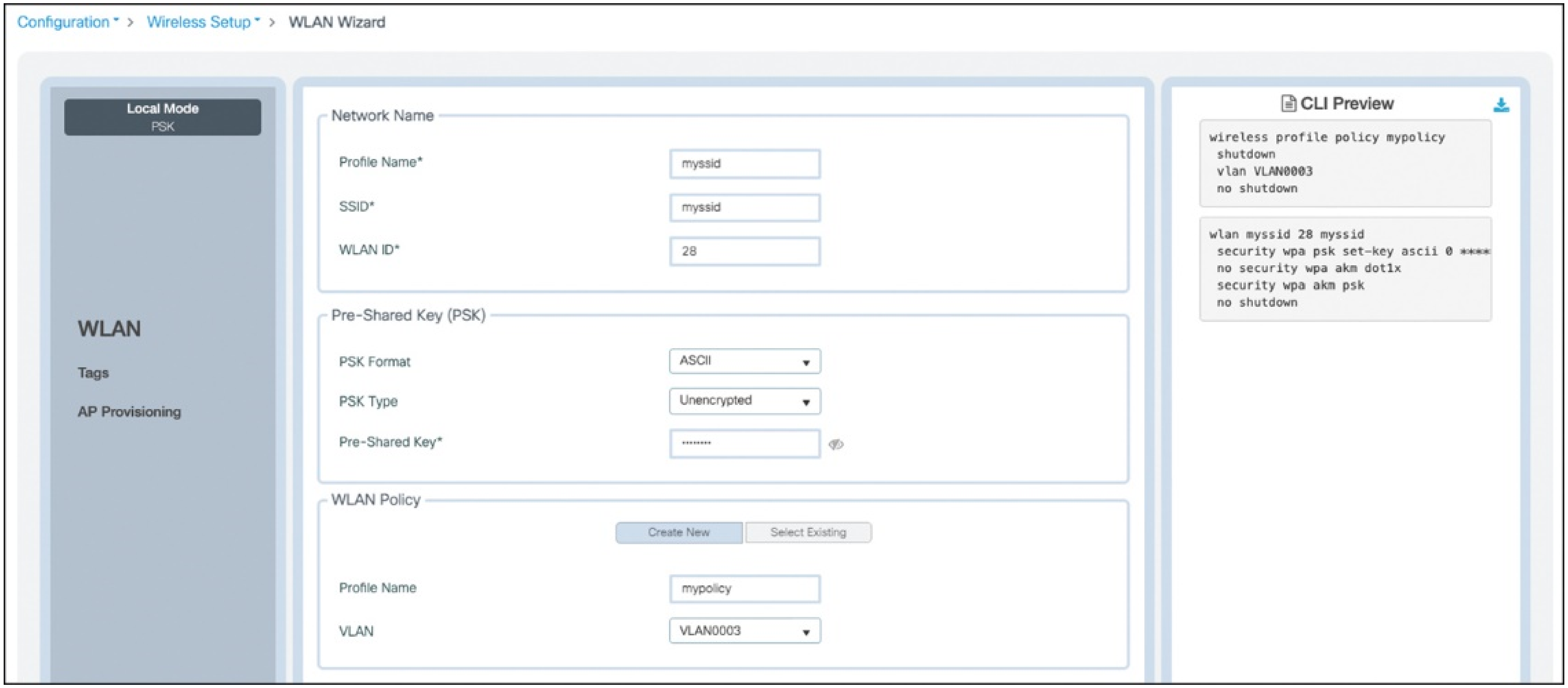


Figure 1-8 Setting up a PSK SSID with the WLAN Wizard

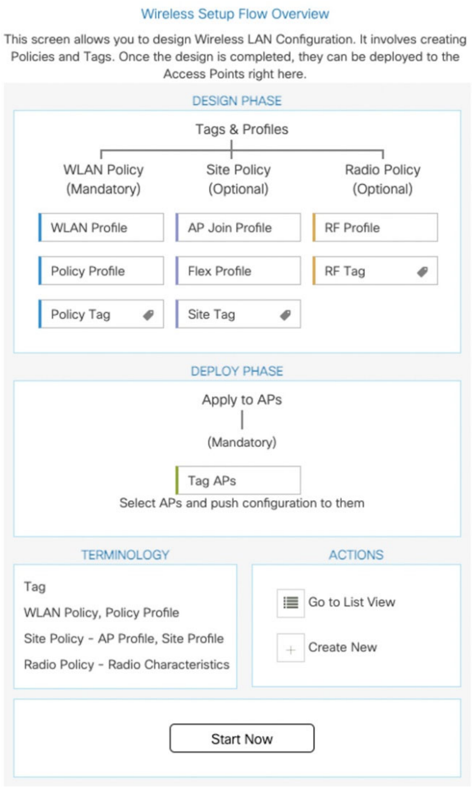


Figure 1-9 The Advanced Setup Wizard

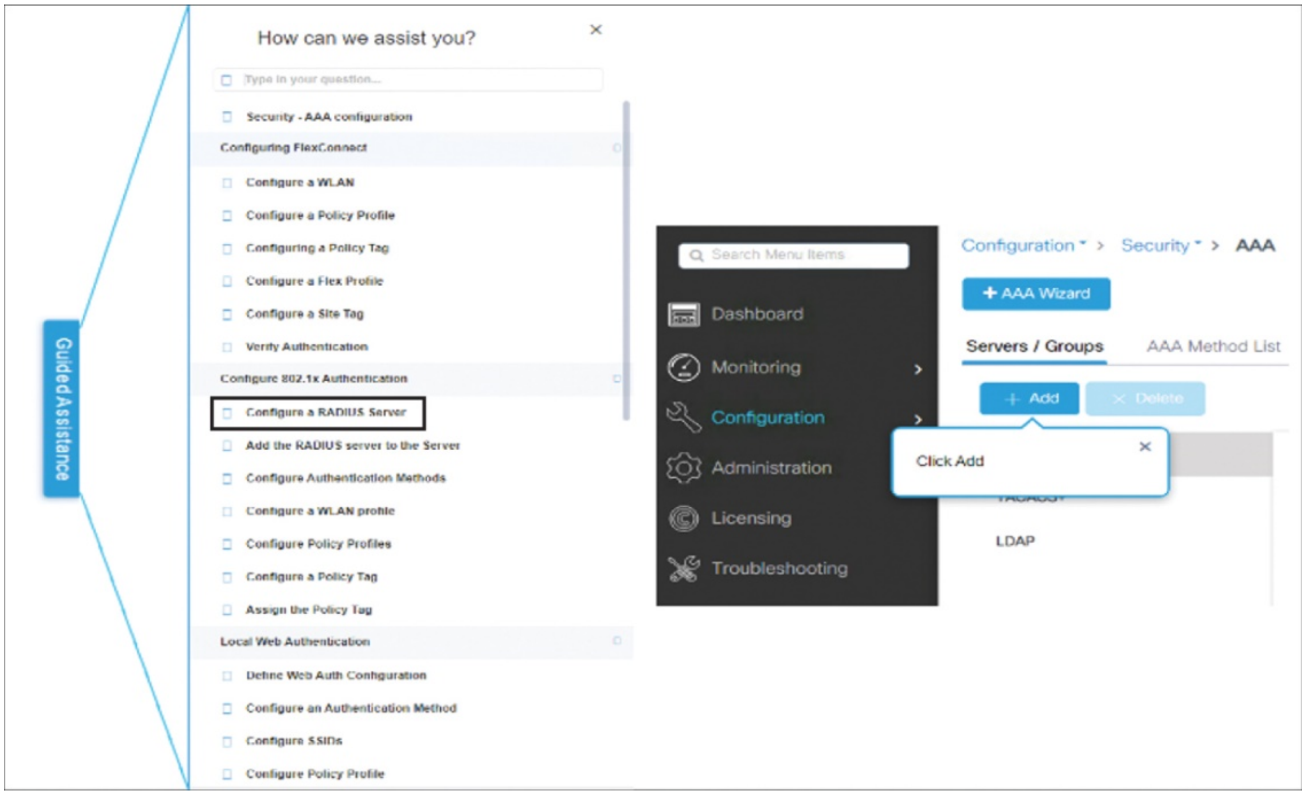


Figure 1-10 Guided Assistance tool

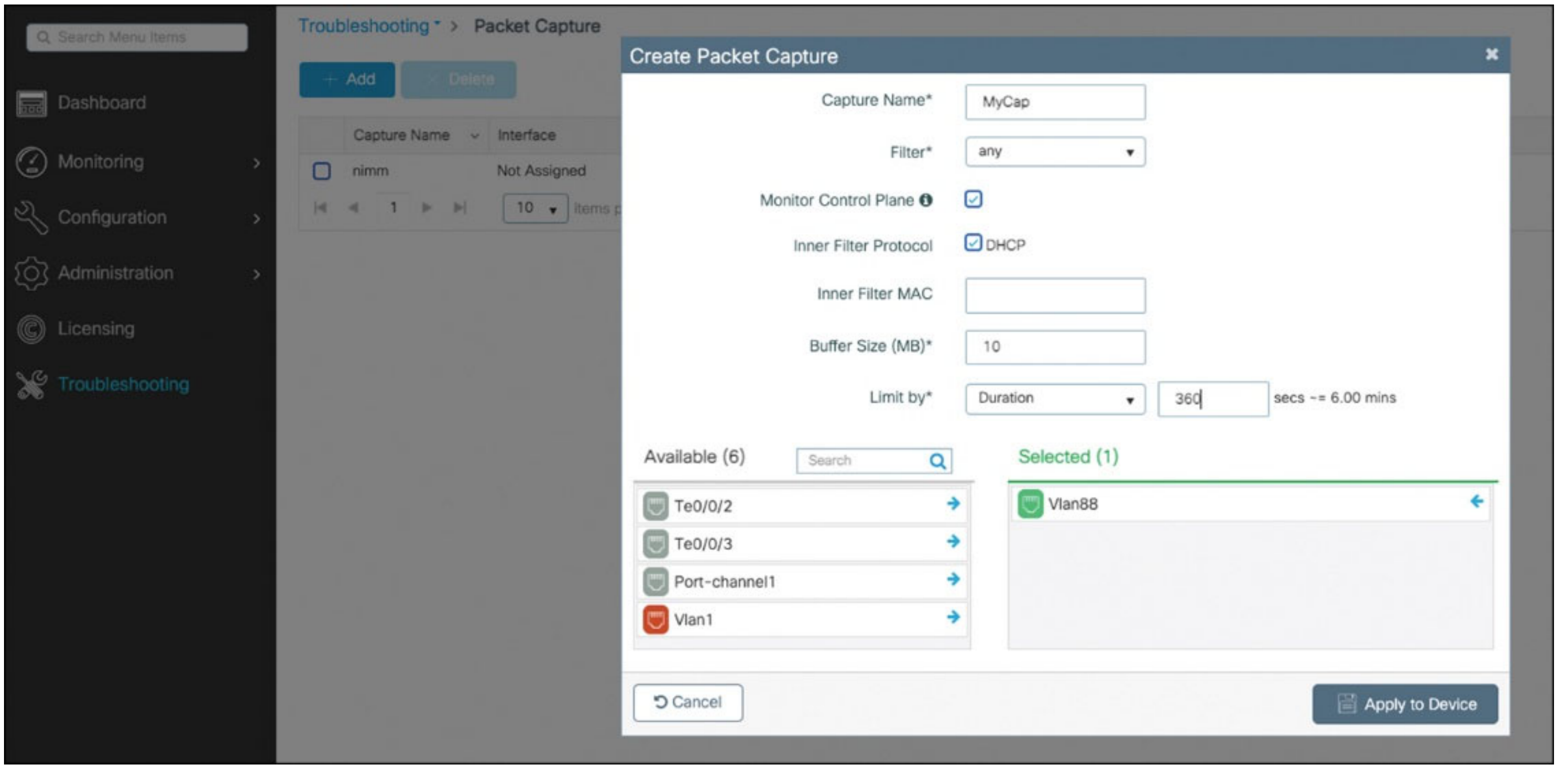


Figure 1-11 Defining a Packet Capture on C9800's GUI

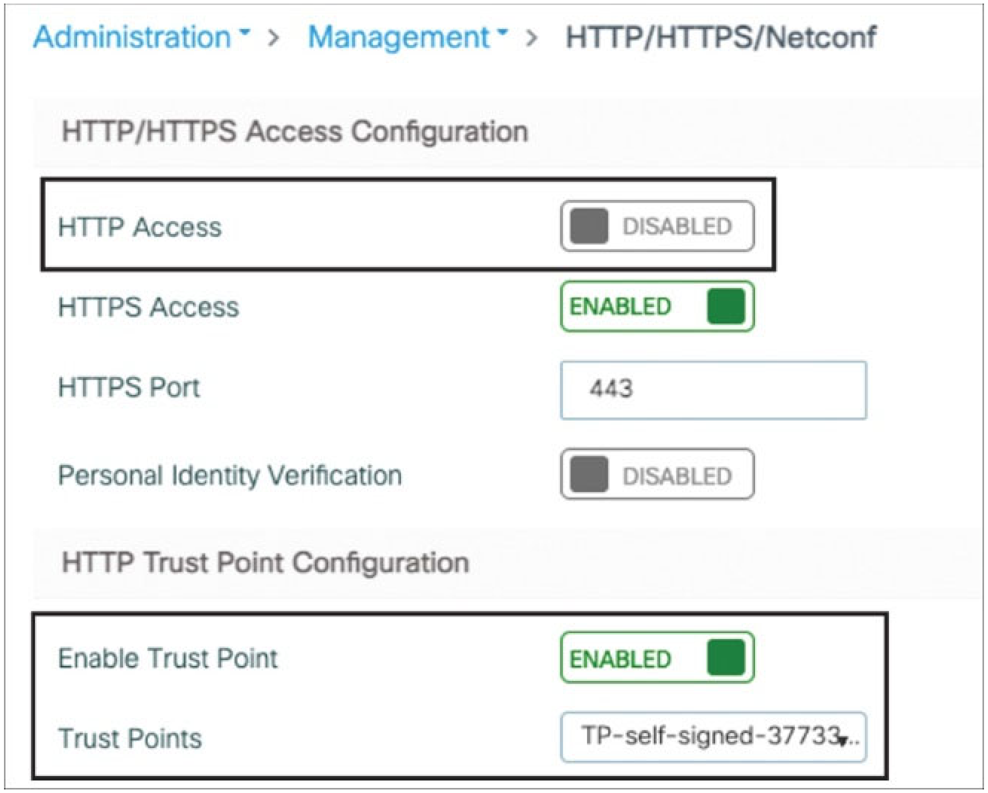


Figure 1-12 Disabling HTTP web access and selecting a specific trustpoint for HTTPs

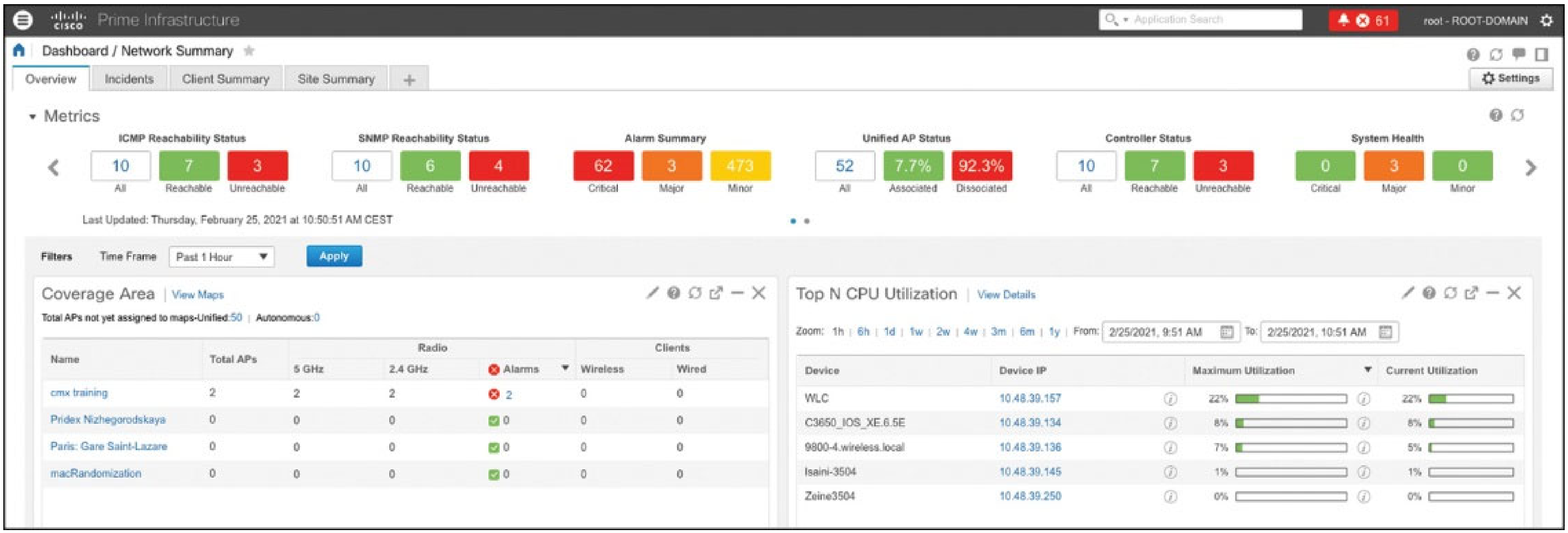


Figure 1-13 Prime Infrastructure dashboard

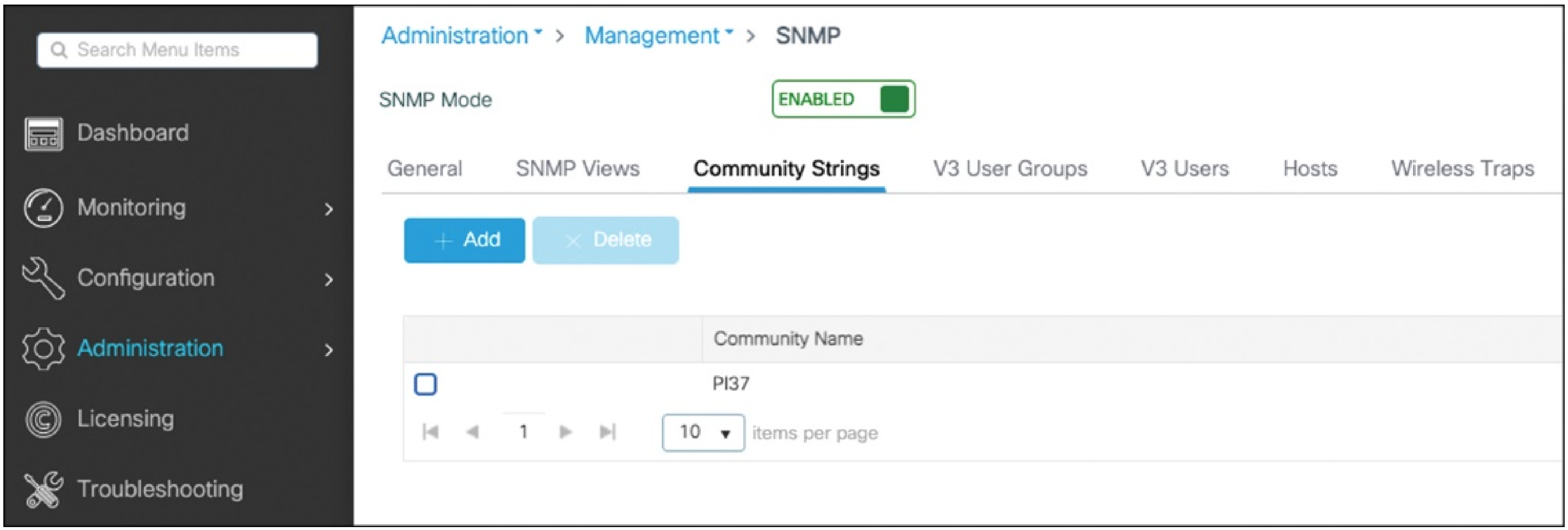


Figure 1-14 SNMP configuration of the C9800

From *Understanding and Troubleshooting Cisco Catalyst 9800 Series Wireless Controller by* Simone Arena, Francisco Sedano Crippa, Nicolas Darchis and Sudha Katgeri(9780137492329) Cisco Press

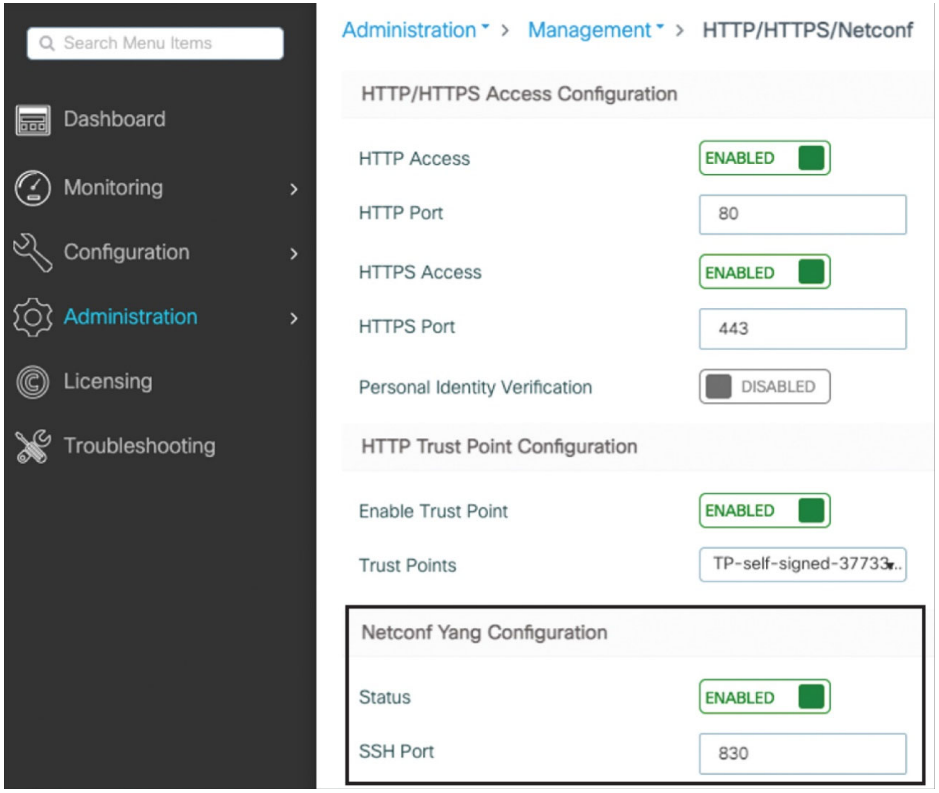


Figure 1-15 Enabling NETCONF on the Catalyst 9800 GUI

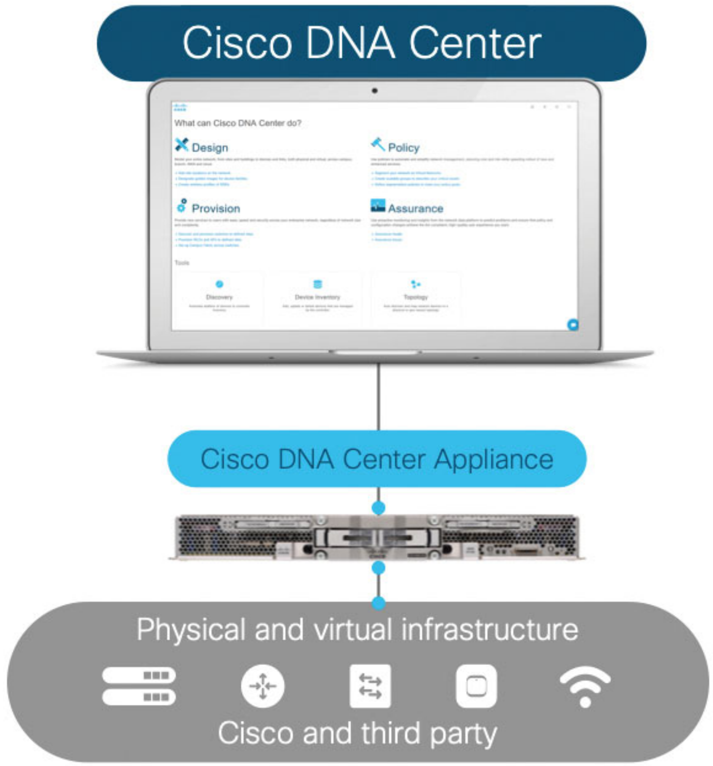


Figure 1-16 Cisco DNA Center

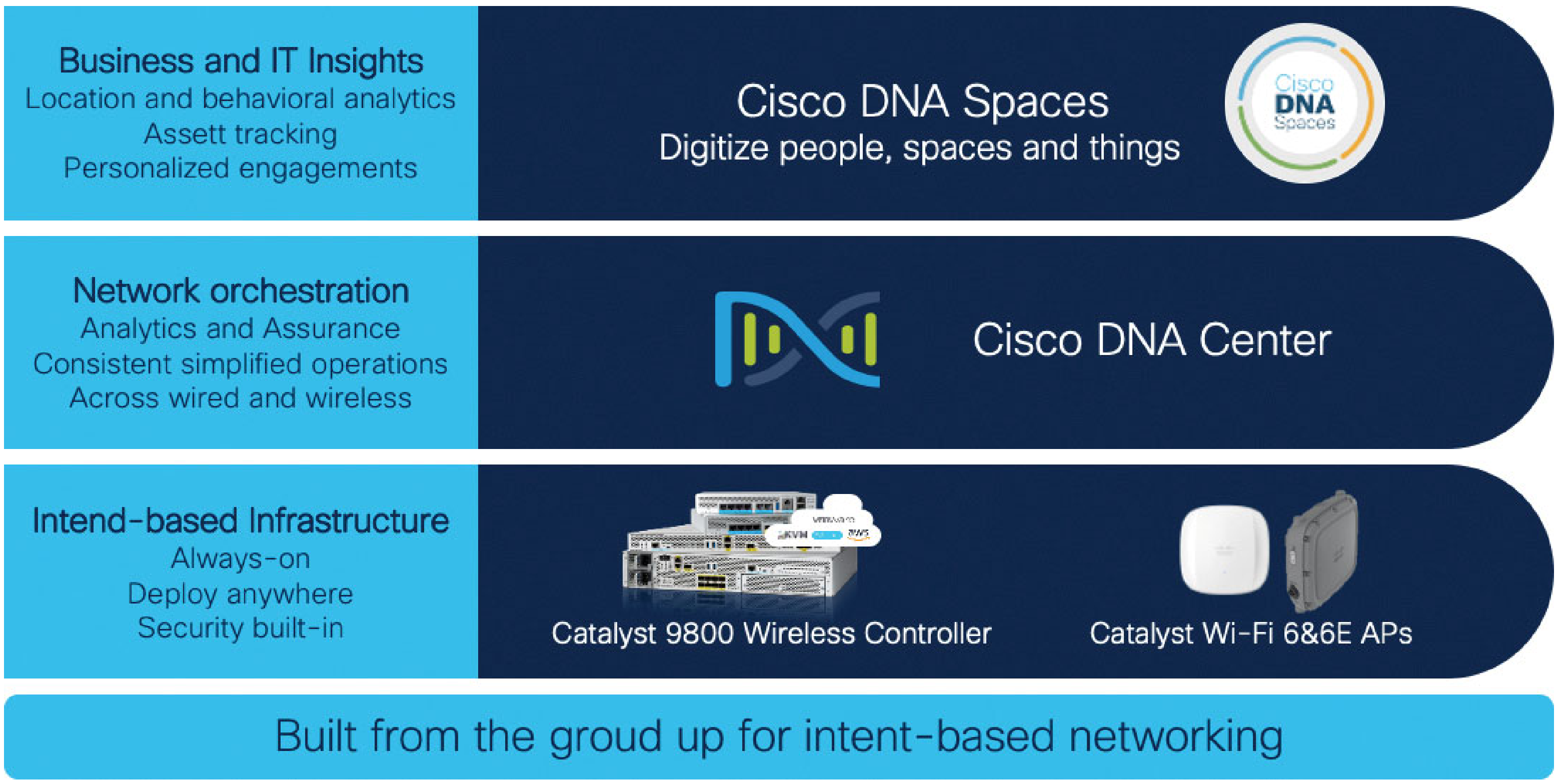


Figure 1-17 Cisco next-generation wireless stack

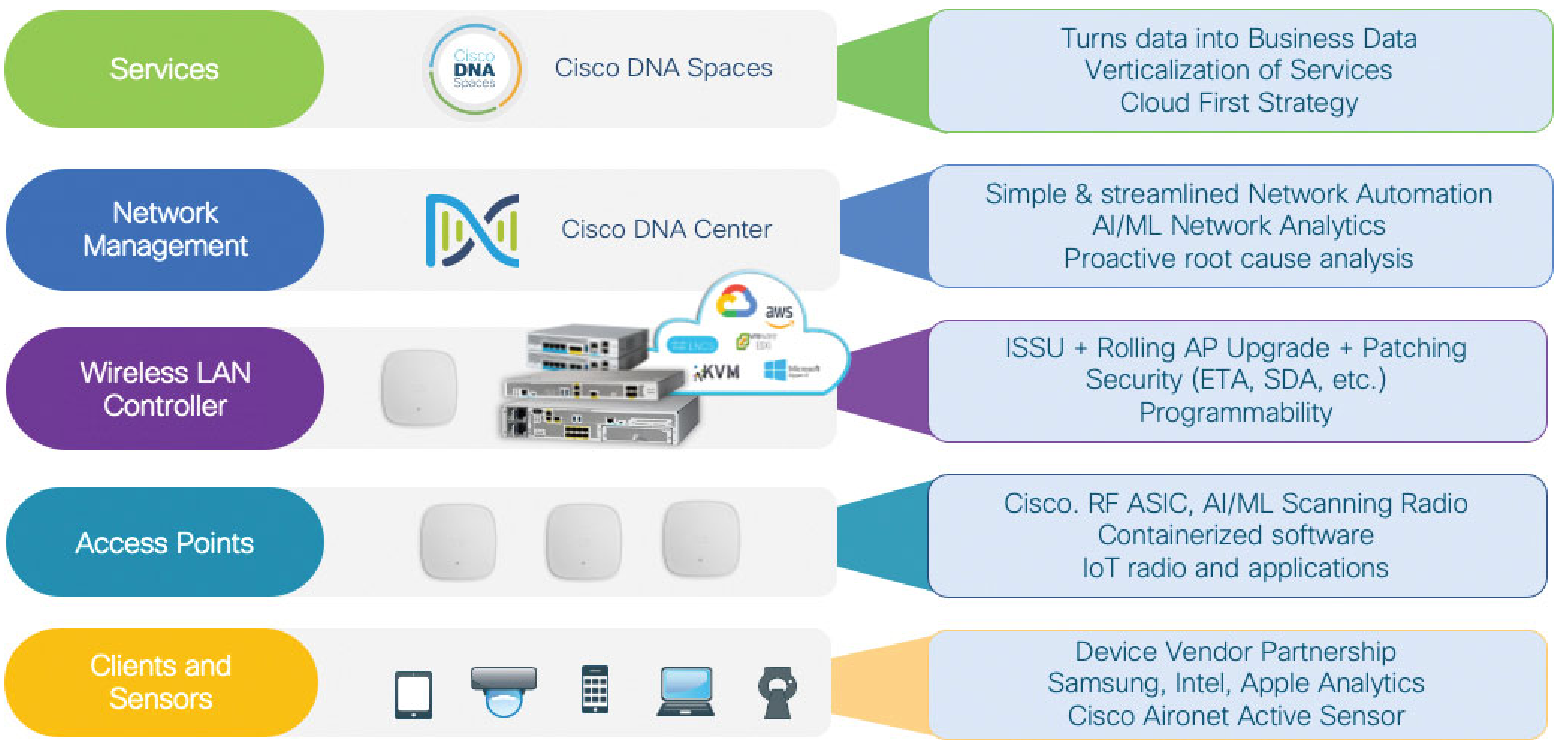


Figure 1-18 Wireless innovation at each later of the stack

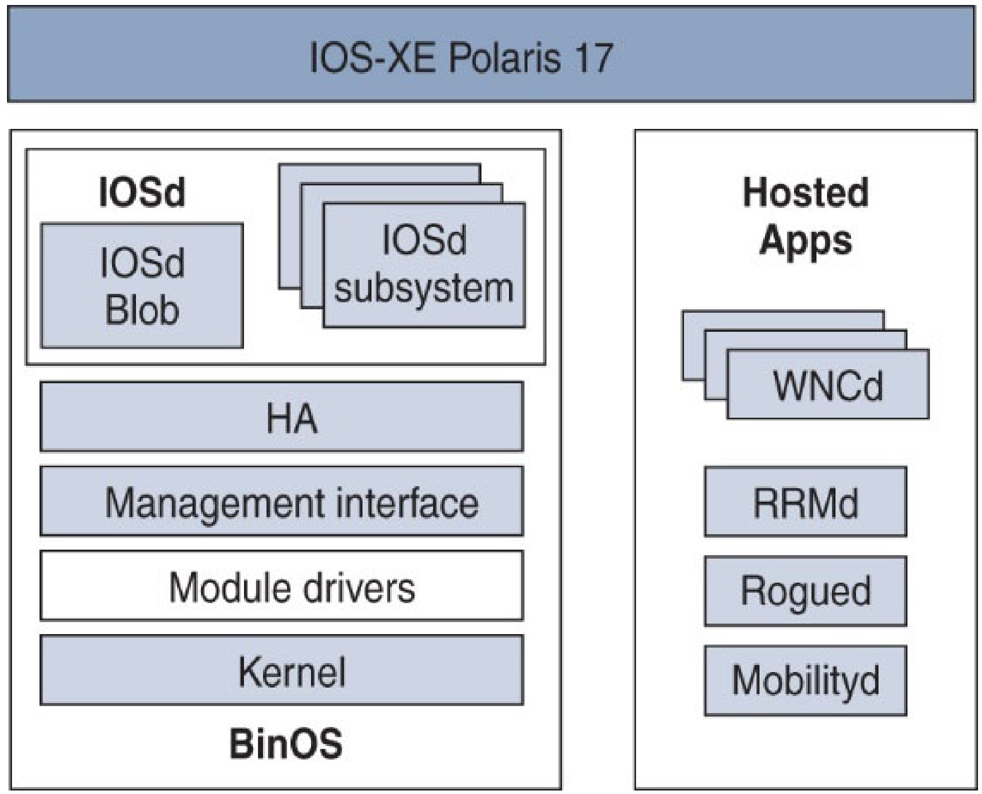


Figure 2-1 IOS-XE general software architecture

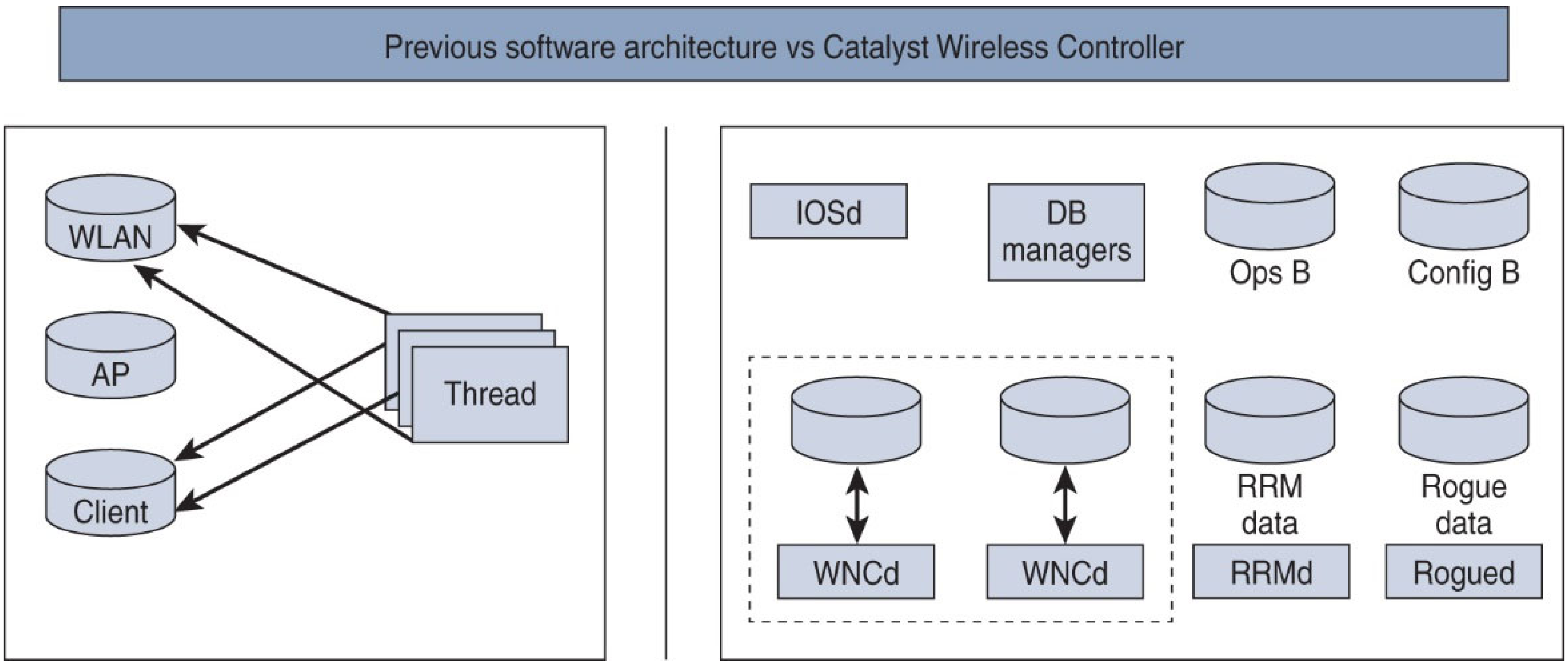


Figure 2-2 The old wireless architecture on the left compared to the Catalyst Wireless architecture on the right

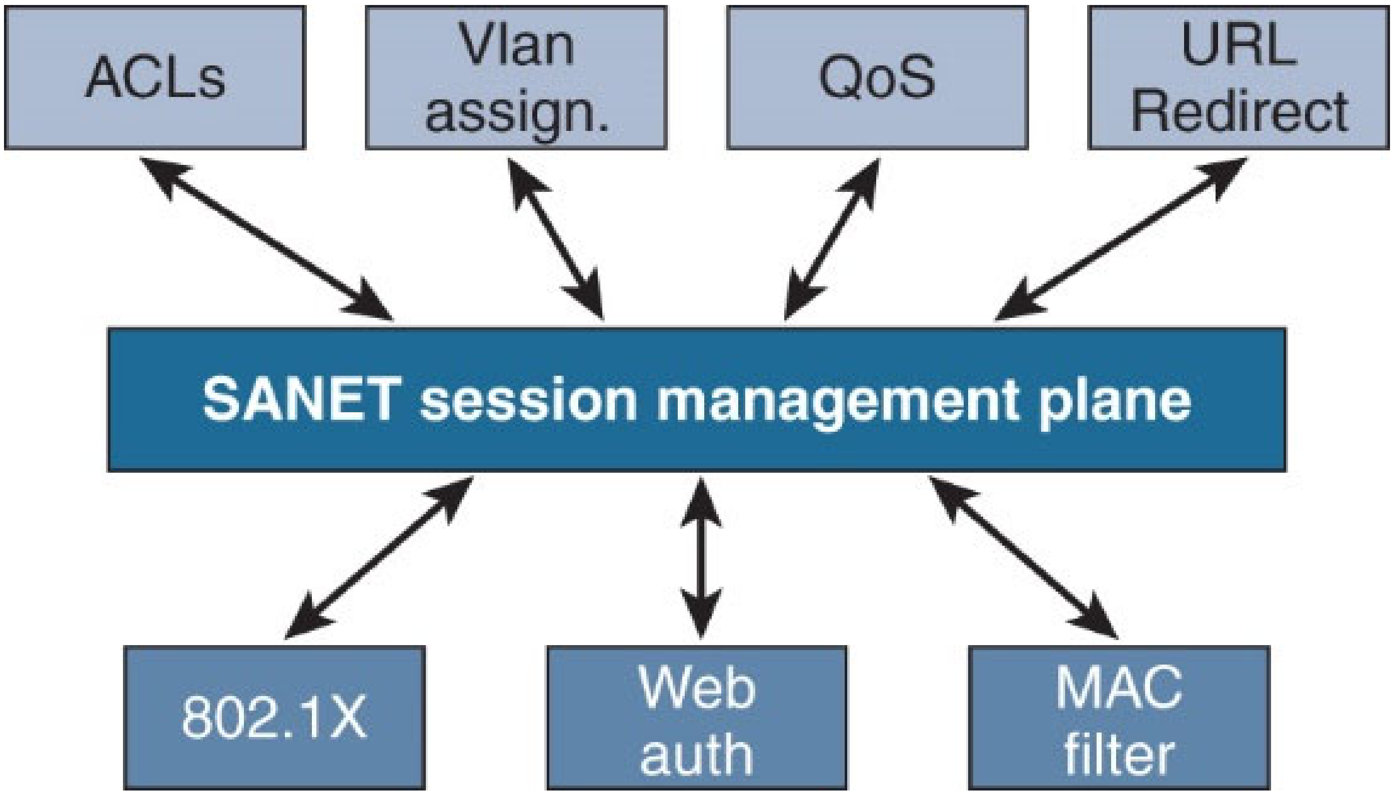


Figure 2-3 SANET library responsibilities

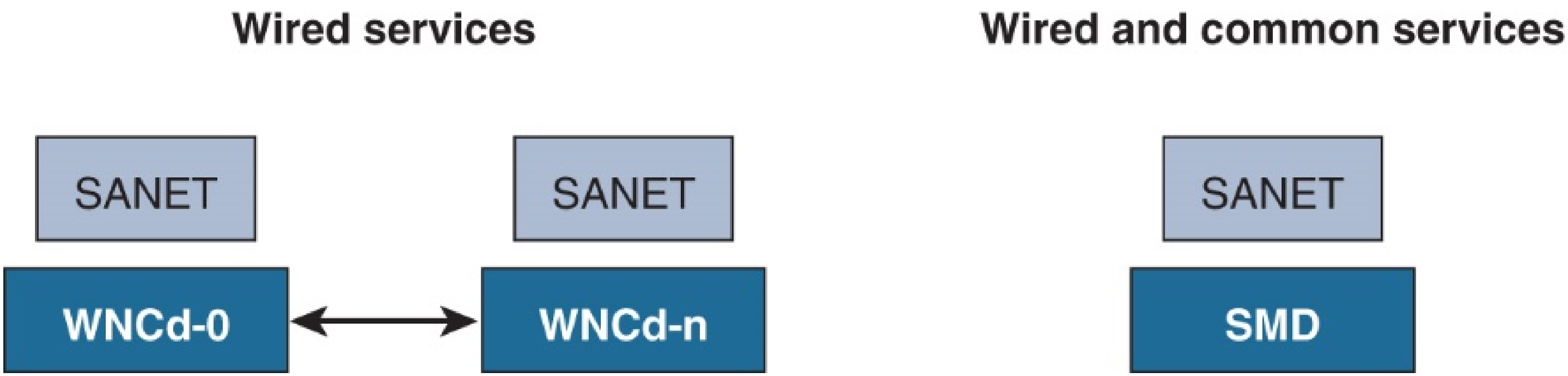


Figure 2-4 WNCd processes and the AAA task split between the existing SMD process and the new SANET libraries inside WNCds

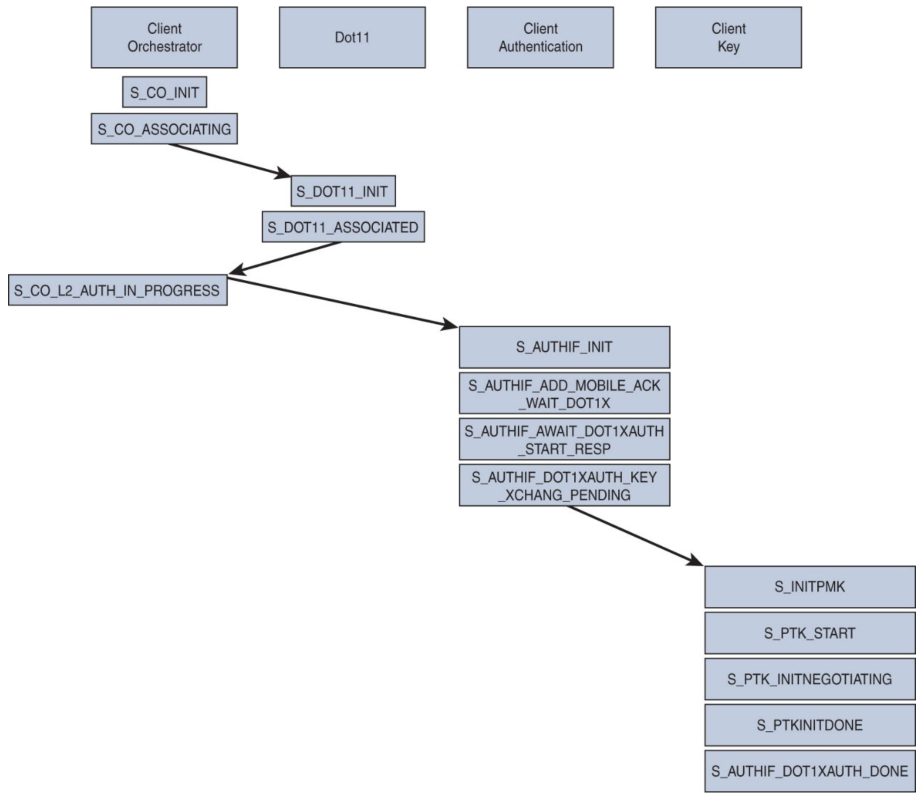


Figure 2-5 Client state machine on a WPA2 Enterprise SSID

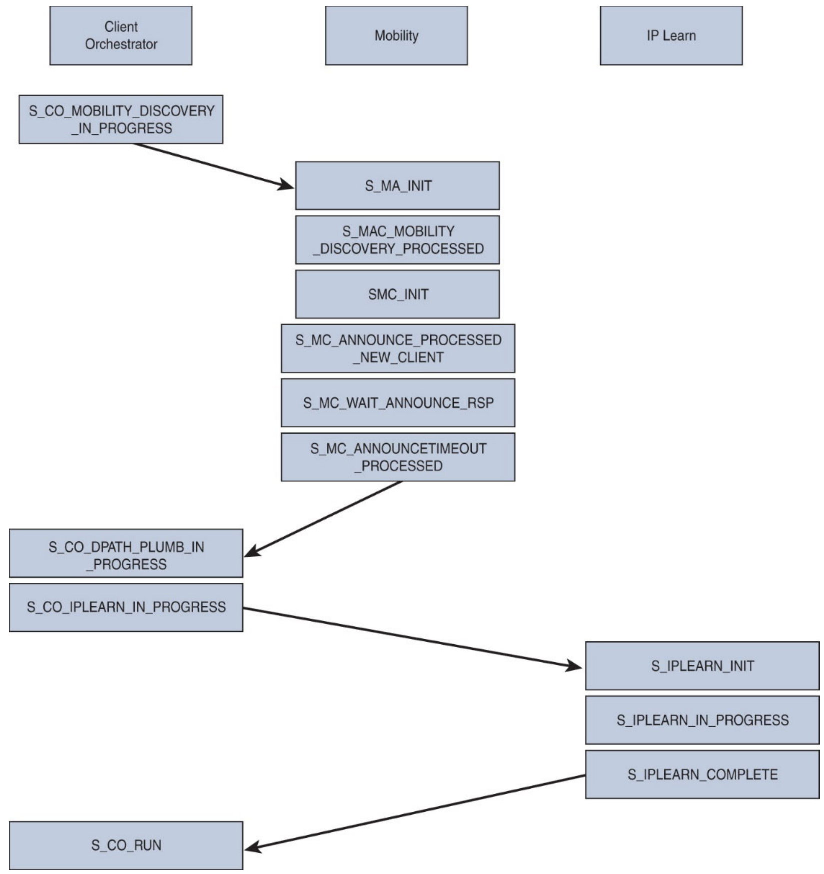


Figure 2-6 Client state machine on a WPA2 Enterprise SSID (continued)

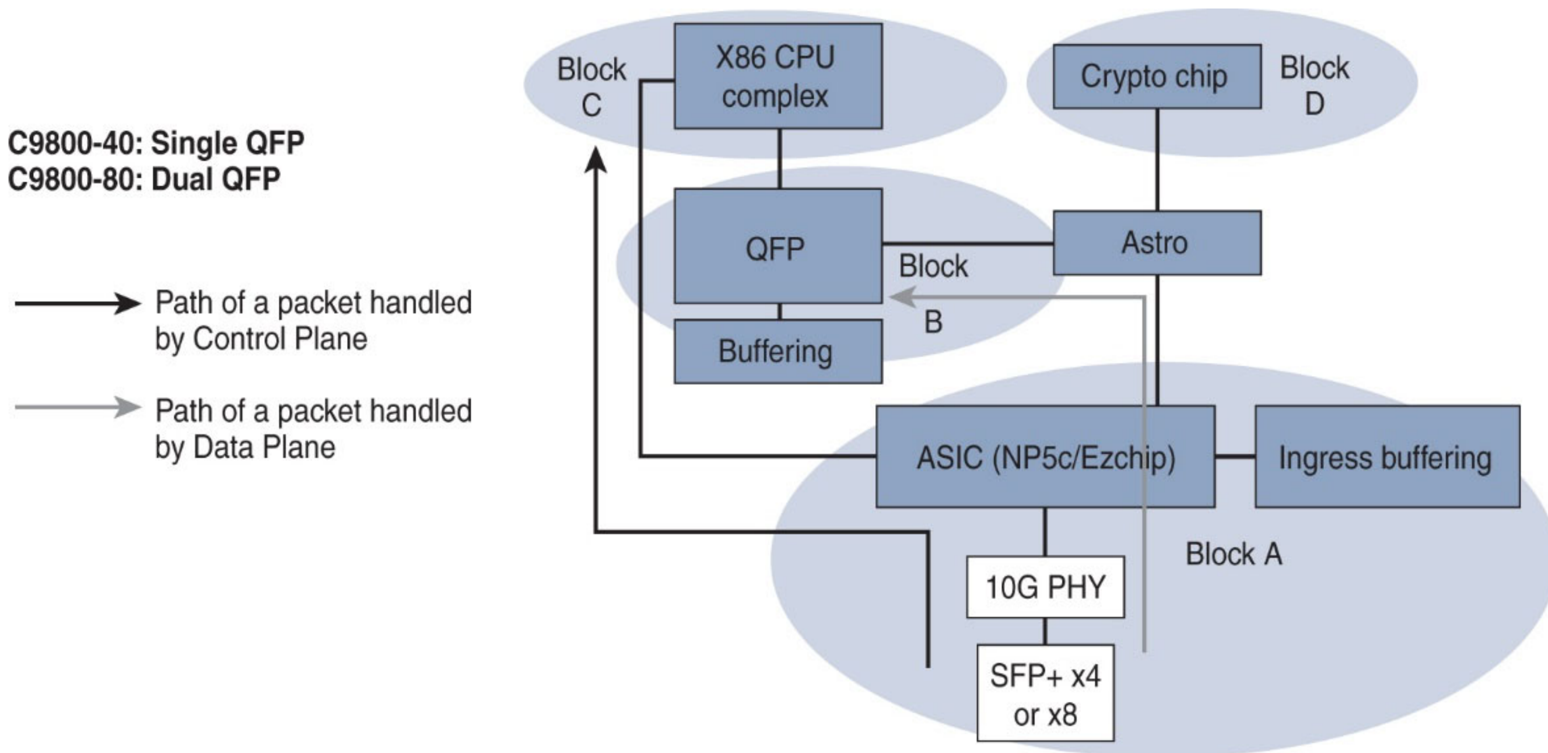


Figure 2-7 Block diagram of the life of a packet in the 9800 appliance's dataplane

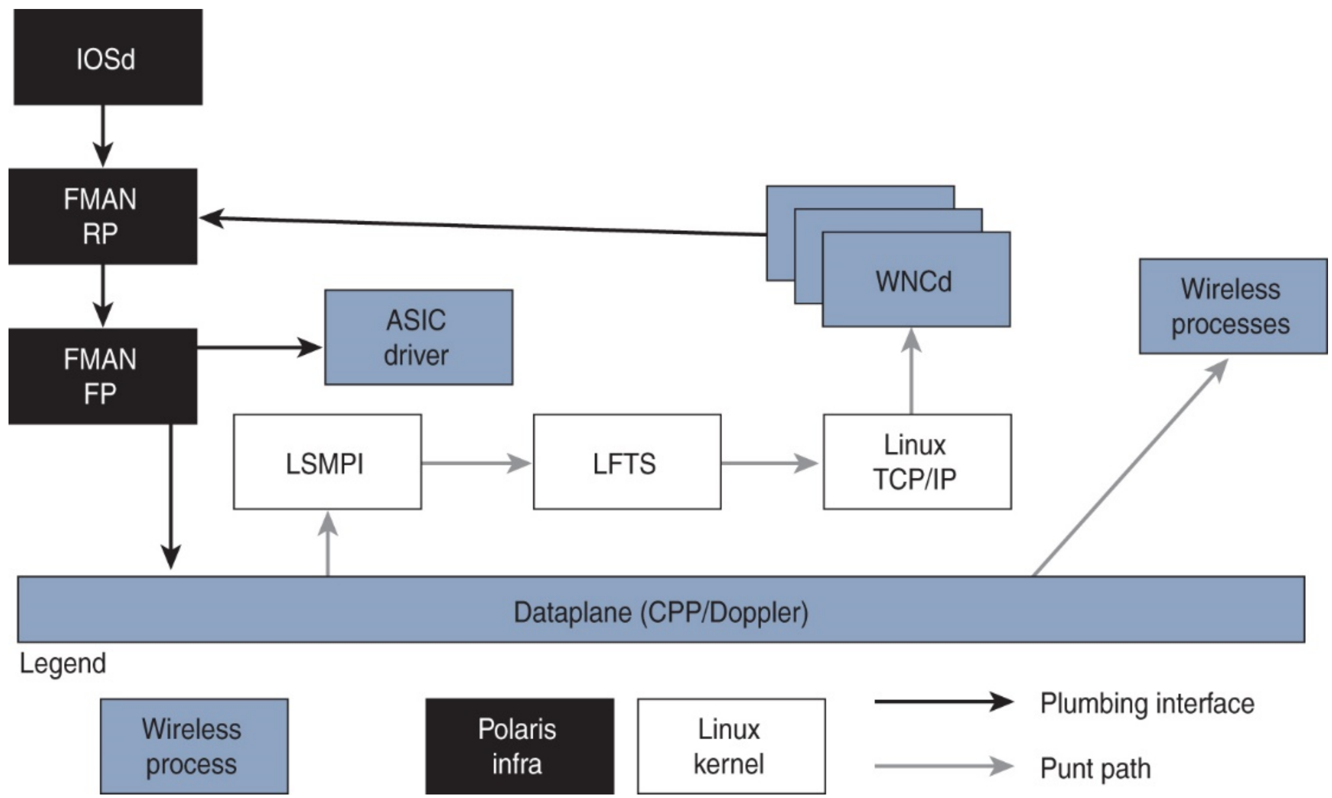


Figure 2-8 Control Plane packet processing

Figure 2-9 A 9800-40 appliance

Figure 2-10 A 9800-80 appliance

Figure 2-11 A 9800-L-C appliance

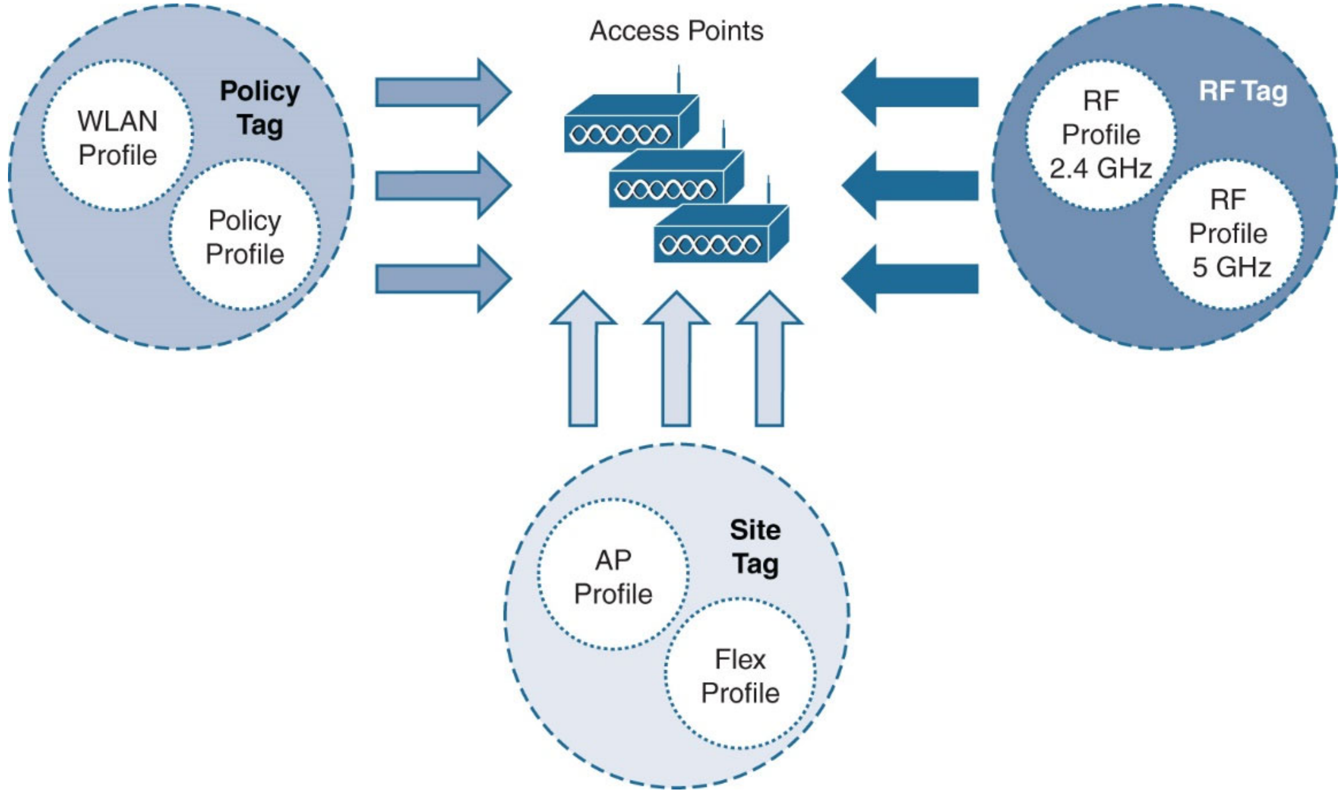


Figure 3-1 Profiles and tags and AP assignment

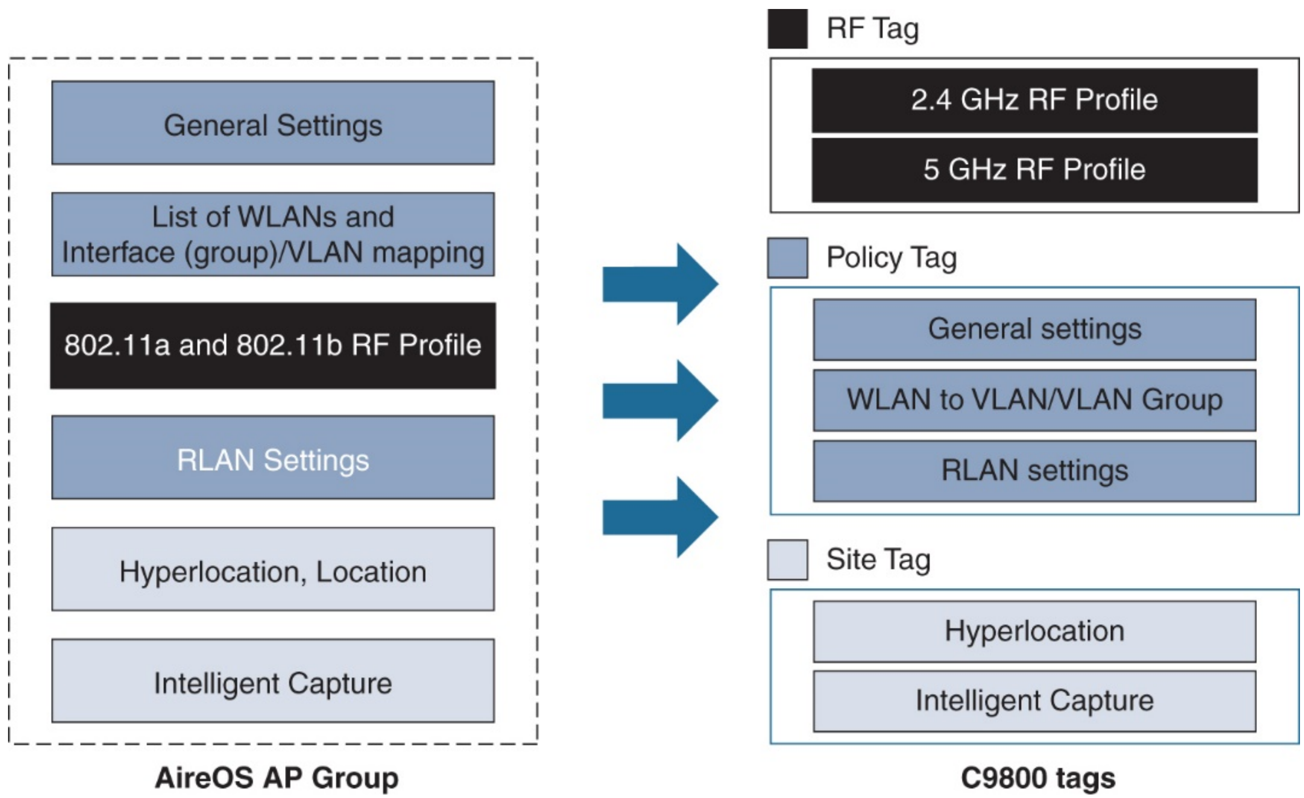


Figure 3-2 AireOS AP Group to C9800 site tags mapping

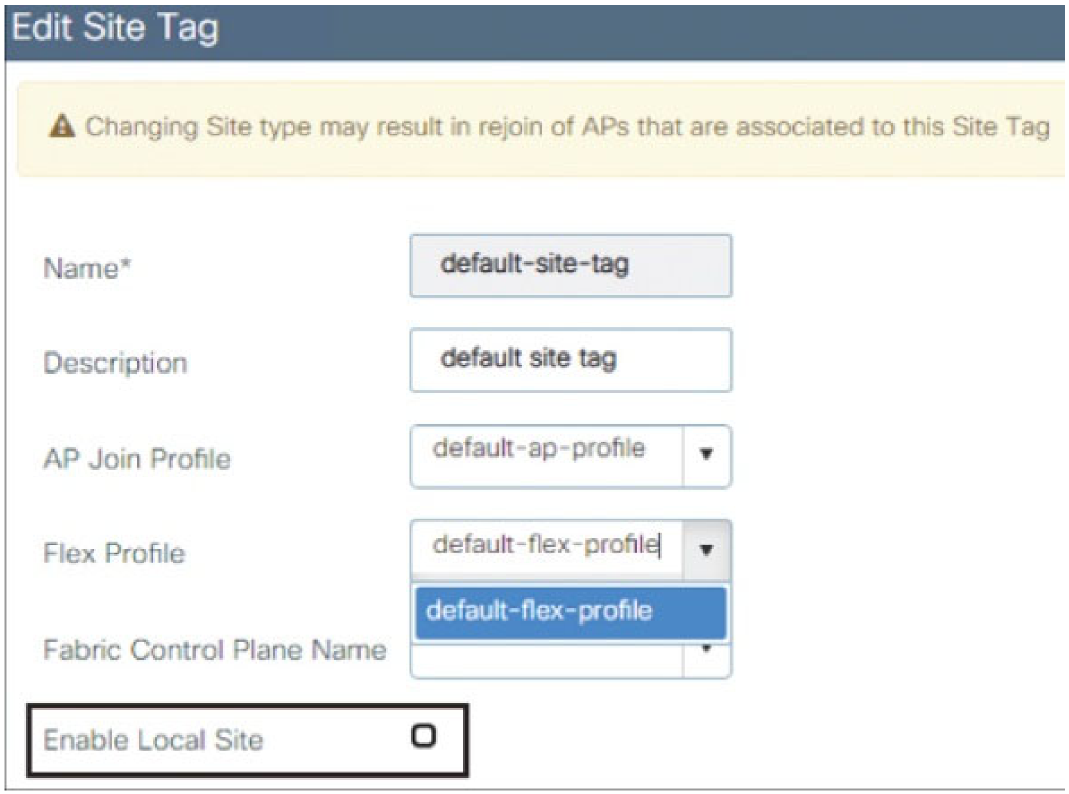


Figure 3-3 Setting the site tag as FlexConnect = nonlocal site

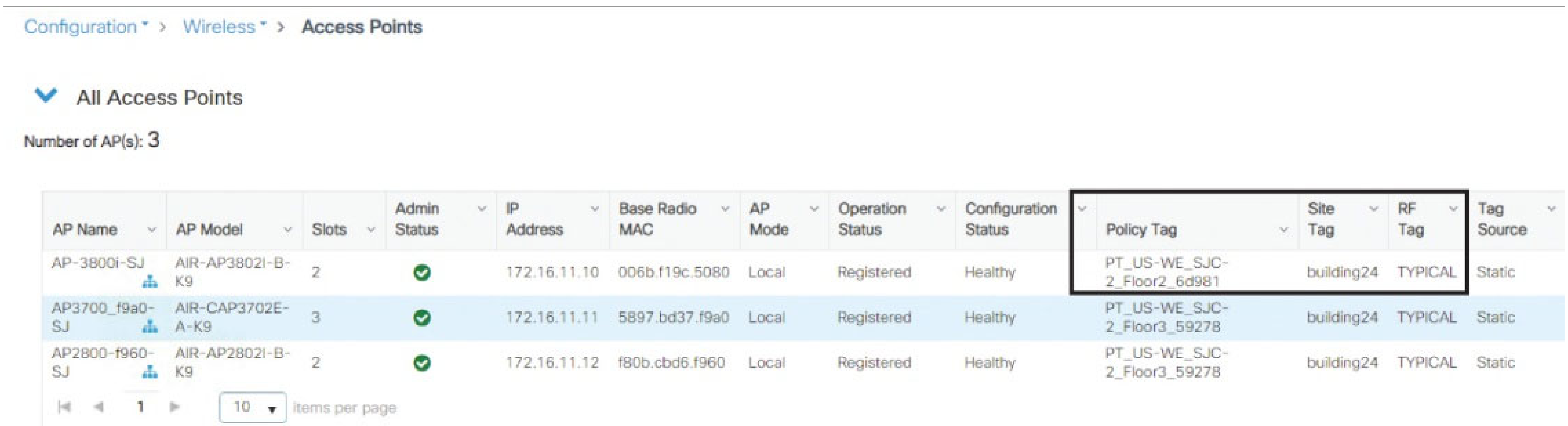


Figure 3-4 AP tags in the GUI

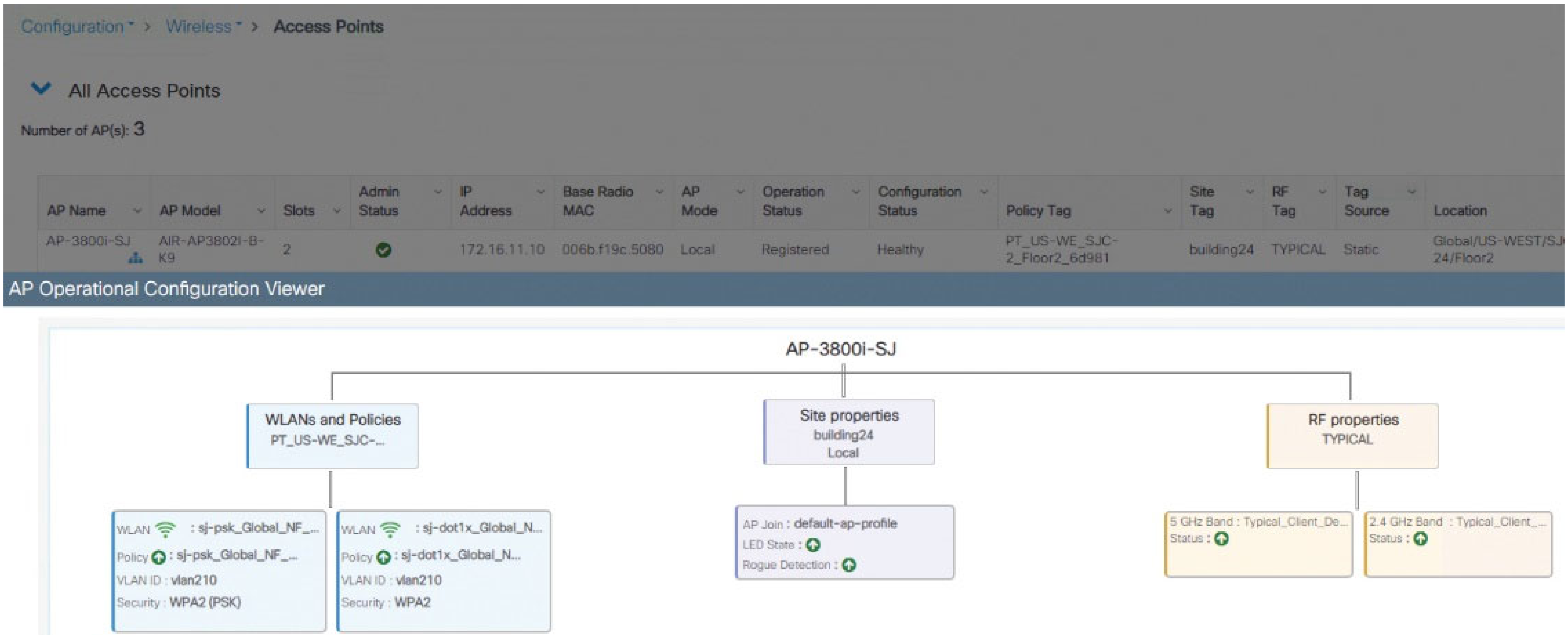


Figure 3-5 AP tags details in the GUI

```
c9800-SJ-11#sh ap tag summary
Number of APs: 3

AP Name            AP Mac           Site Tag Name              Policy Tag Name                 RF Tag Name                  Misconfigured    Tag Source
-------------------------------------------------------------------------------------------------------------------------------------------------------------
AP-3800i-SJ        00a6.ca36.25f2   building24                 PT_US-WE_SJC-2_Floor2_6d981     TYPICAL                      No               Static
AP3700_f9a0-SJ     5897.bd2b.3388   building24                 PT_US-WE_SJC-2_Floor3_59278     TYPICAL                      No               Static
AP2800-f960-SJ     2c33.1180.70fa   building24                 PT_US-WE_SJC-2_Floor3_59278     TYPICAL                      No               Static
```

Figure 3-6 AP tag summary

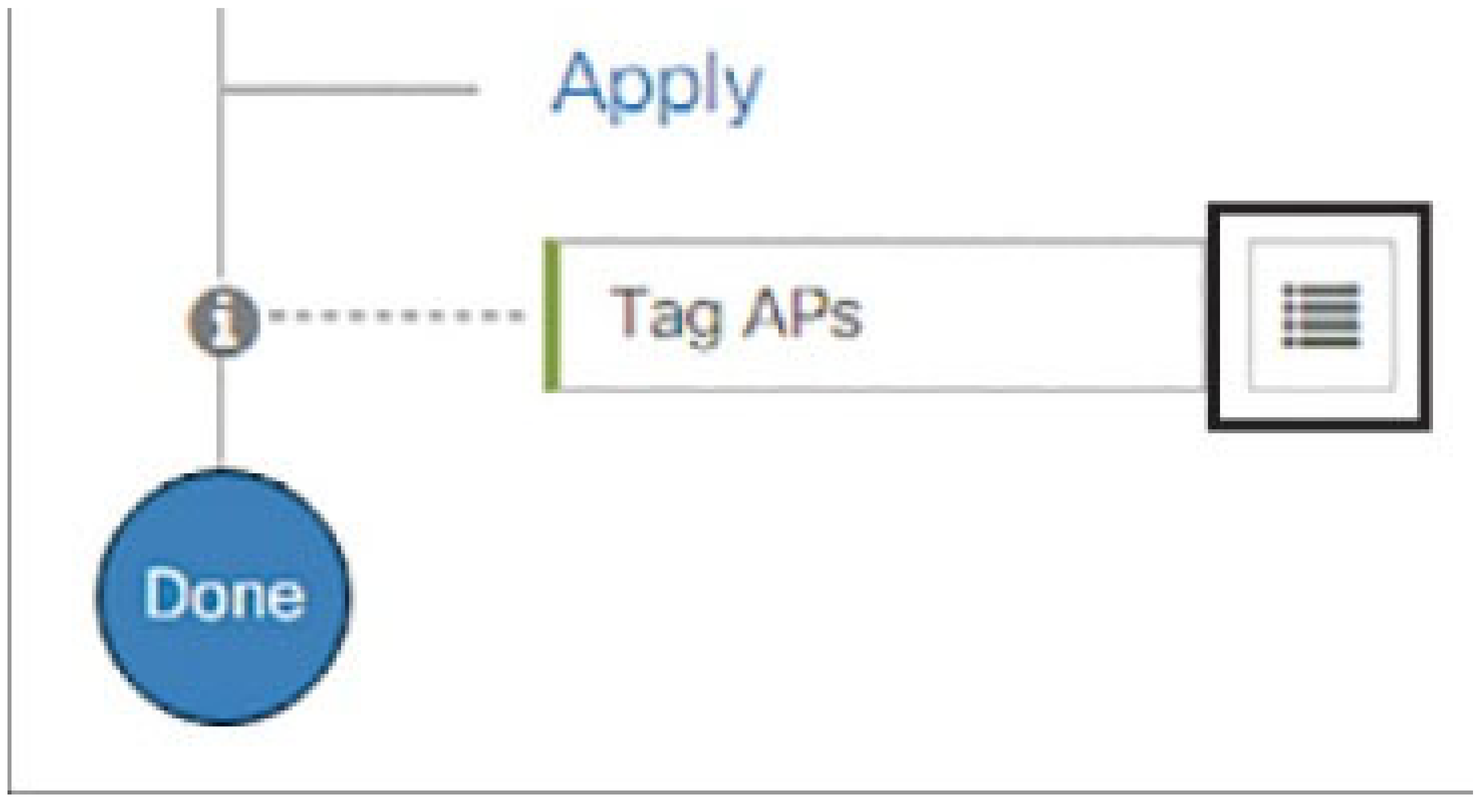


Figure 3-7 Click the menu icon to select APs and assign tags

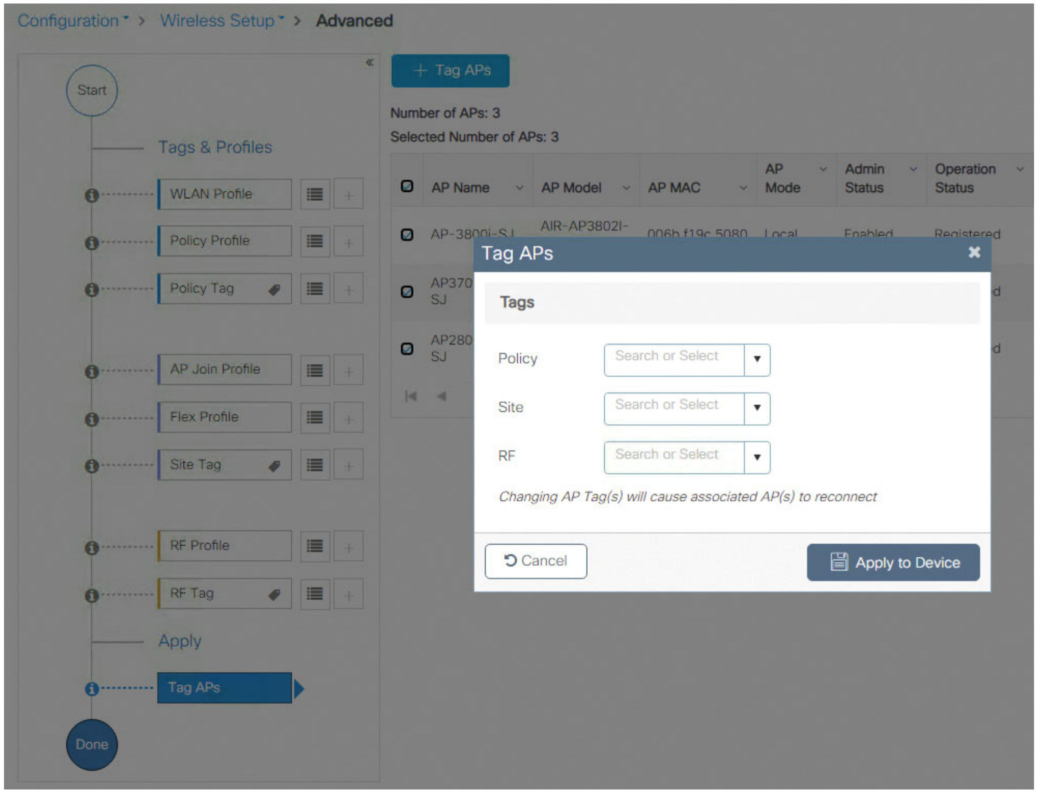


Figure 3-8 Assign the tags to the selected APs

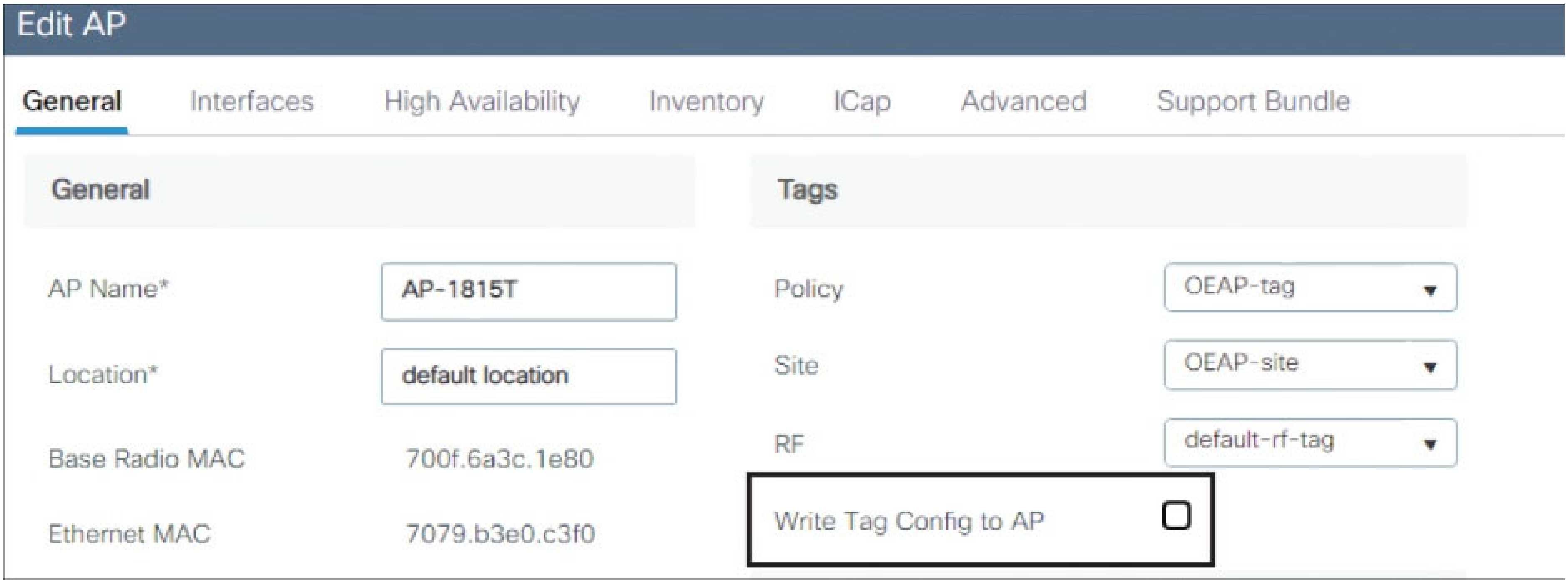


Figure 3-9 Starting with release 17.4.1, the write tag is available in the GUI

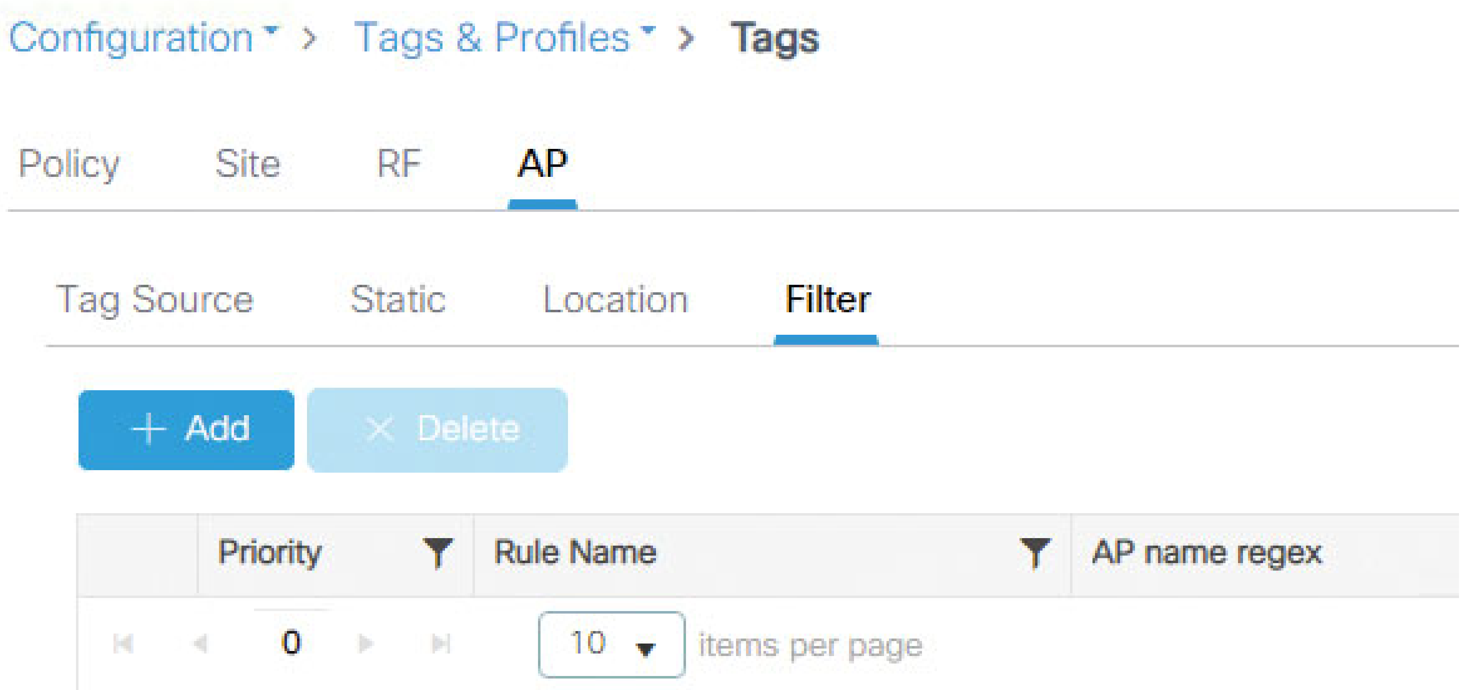


Figure 3-10 Go to the Filter tab to define the tag filter

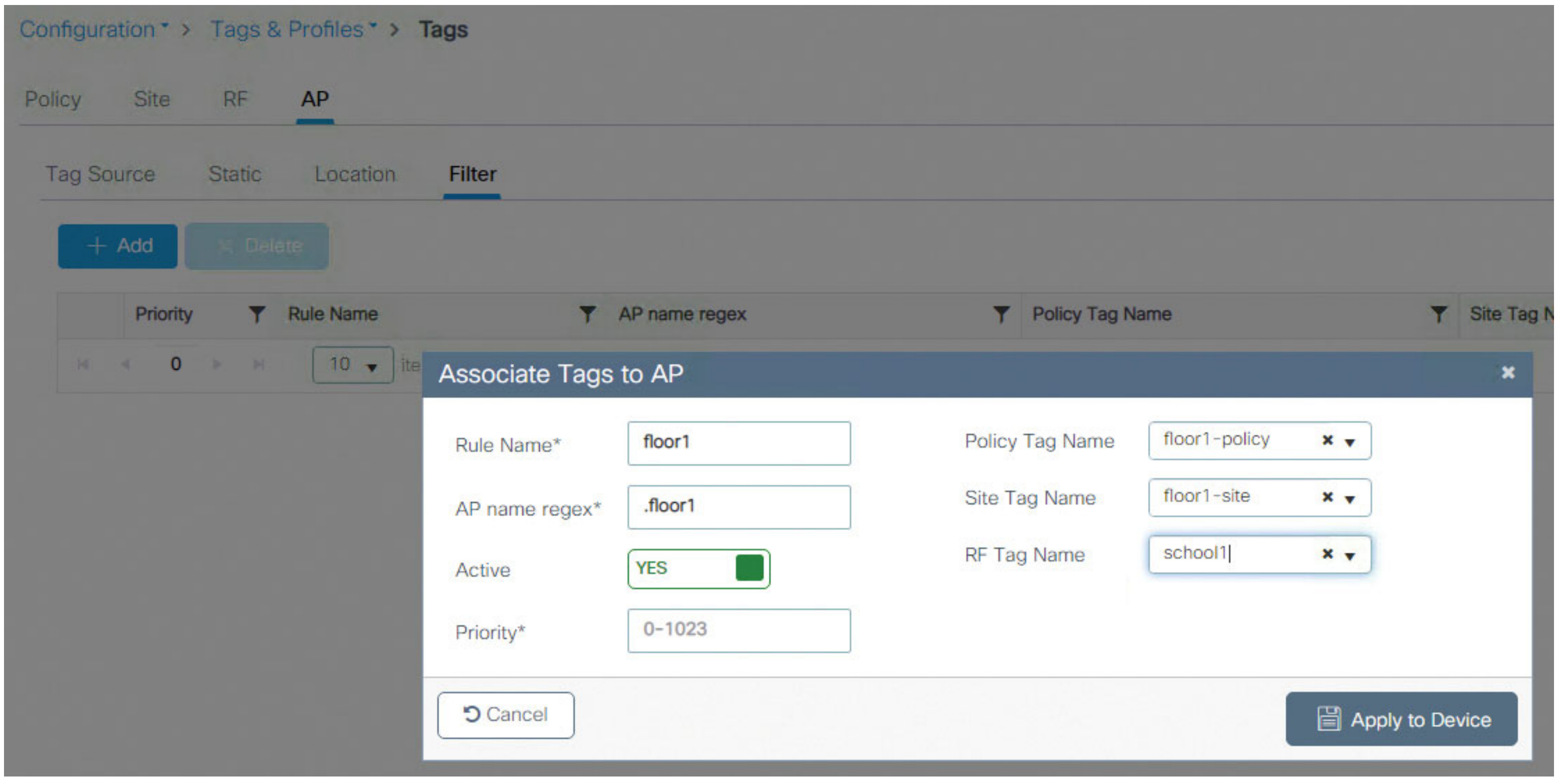


Figure 3-11 A tag filter

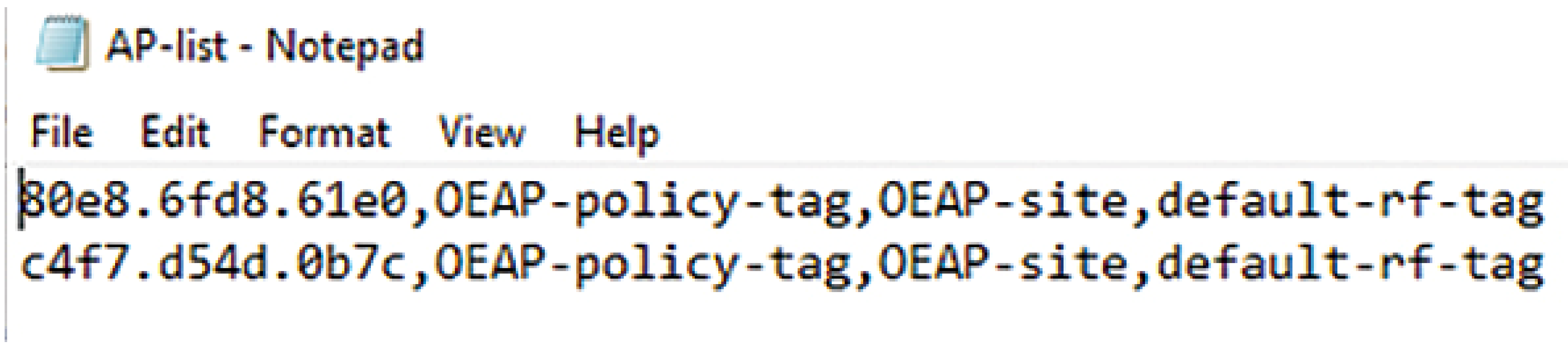


Figure 3-12 CSV import file format example

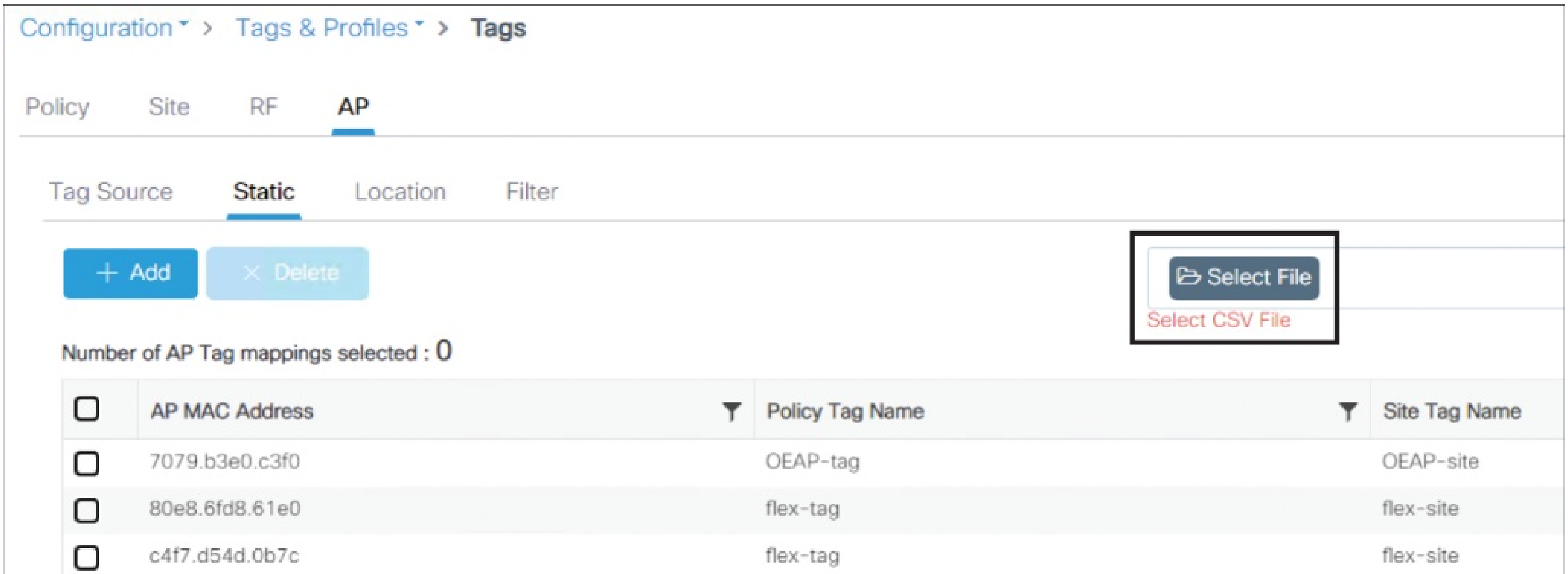


Figure 3-13 Selecting the file to load the AP to tag mapping

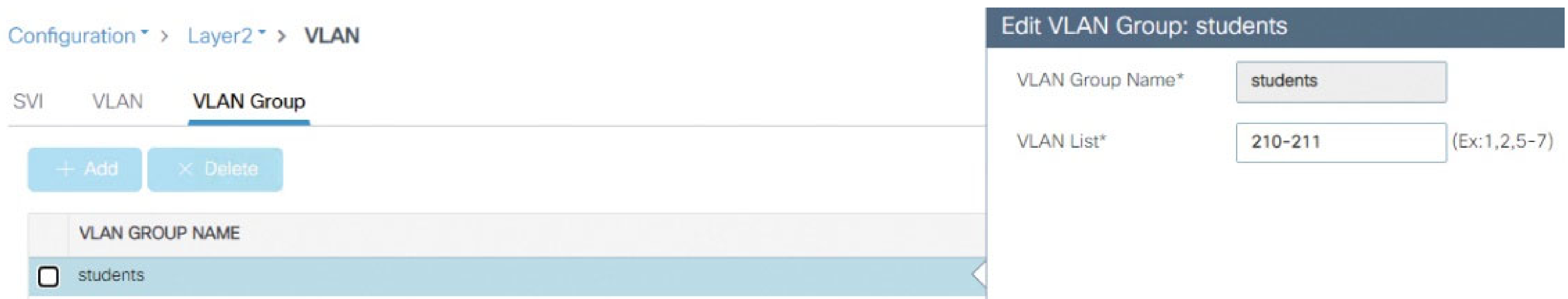


Figure 3-14 VLAN group definition

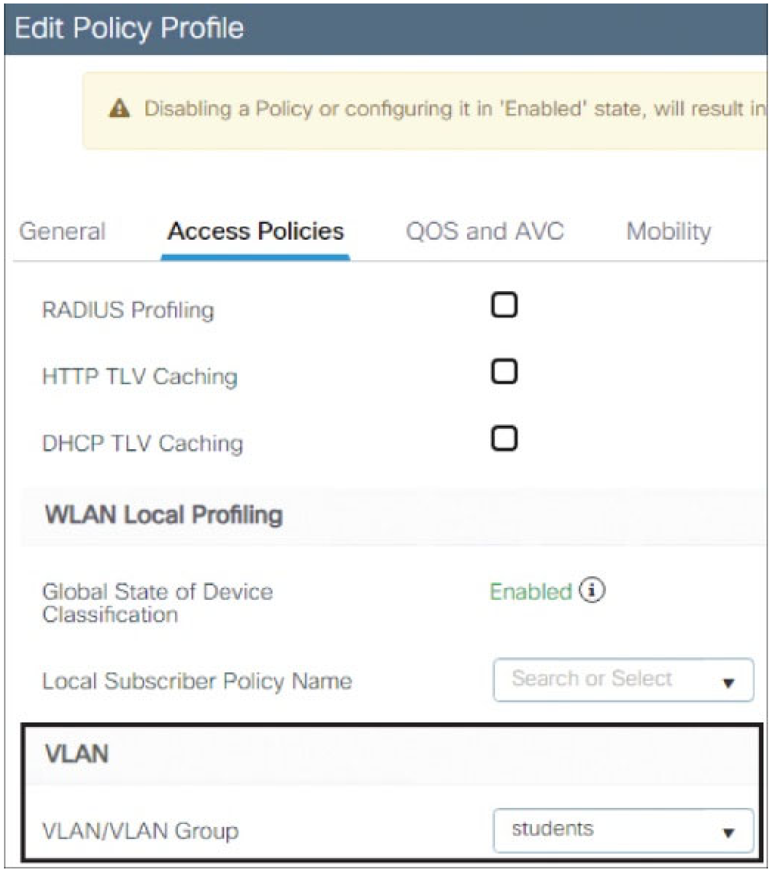


Figure 3-15 VLAN group assignment within the policy profile

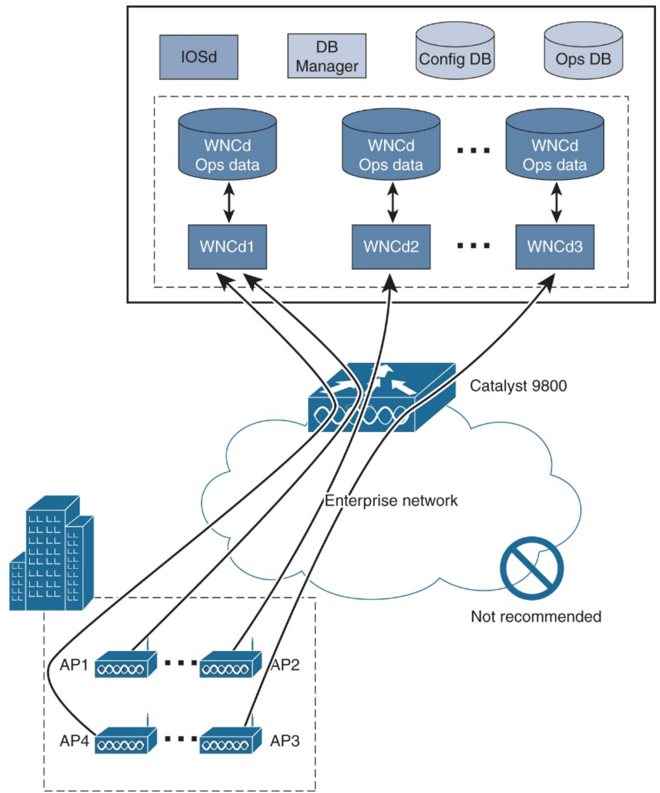


Figure 3-16 APs to WNCd distribution using the default-site-tag

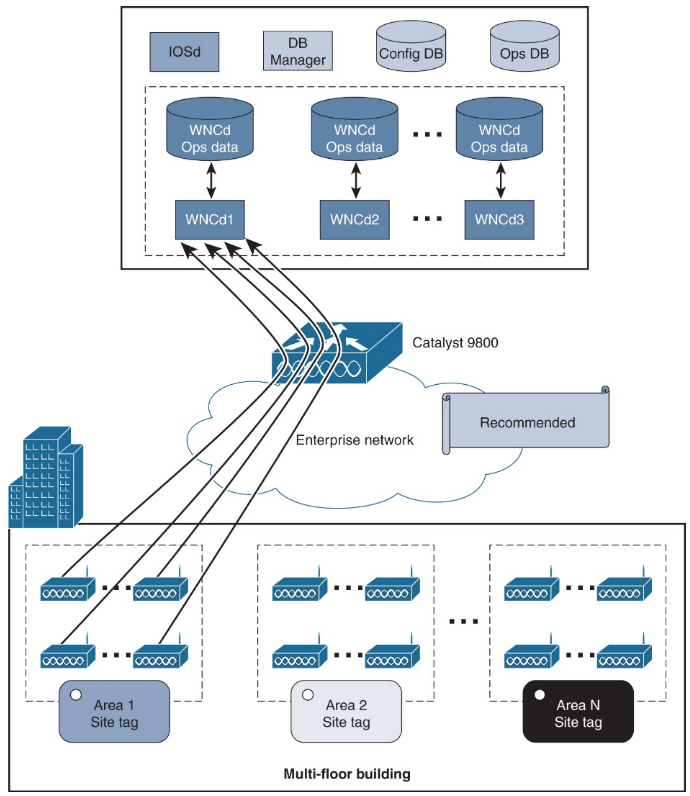


Figure 3-17 APs to WNCd distribution using custom site tag per roaming domain

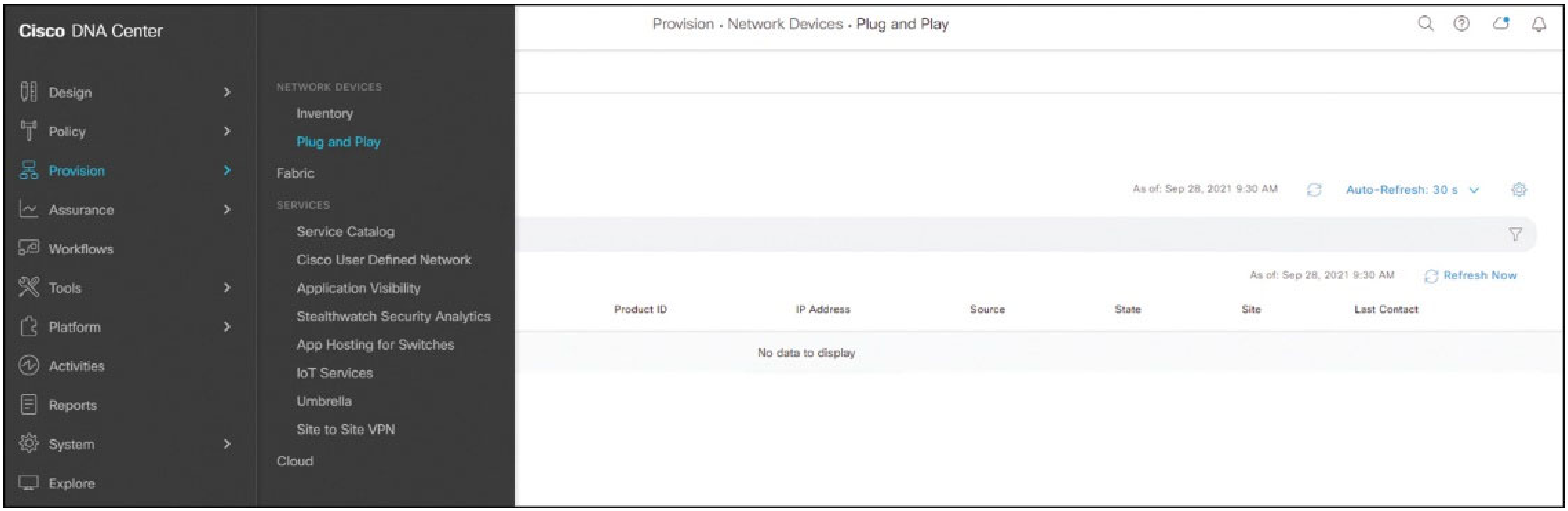


Figure 4-1 Plug and Play provision flow

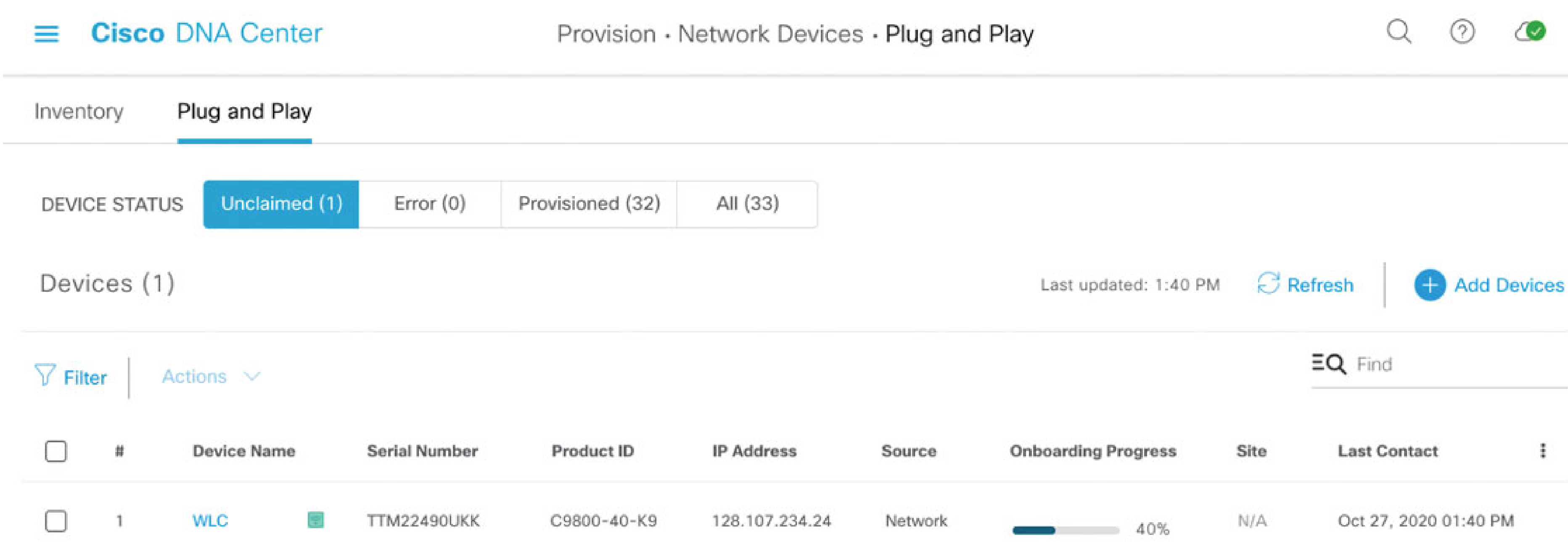


Figure 4-2 C9800 ready to be claimed in Plug and Play dashboard

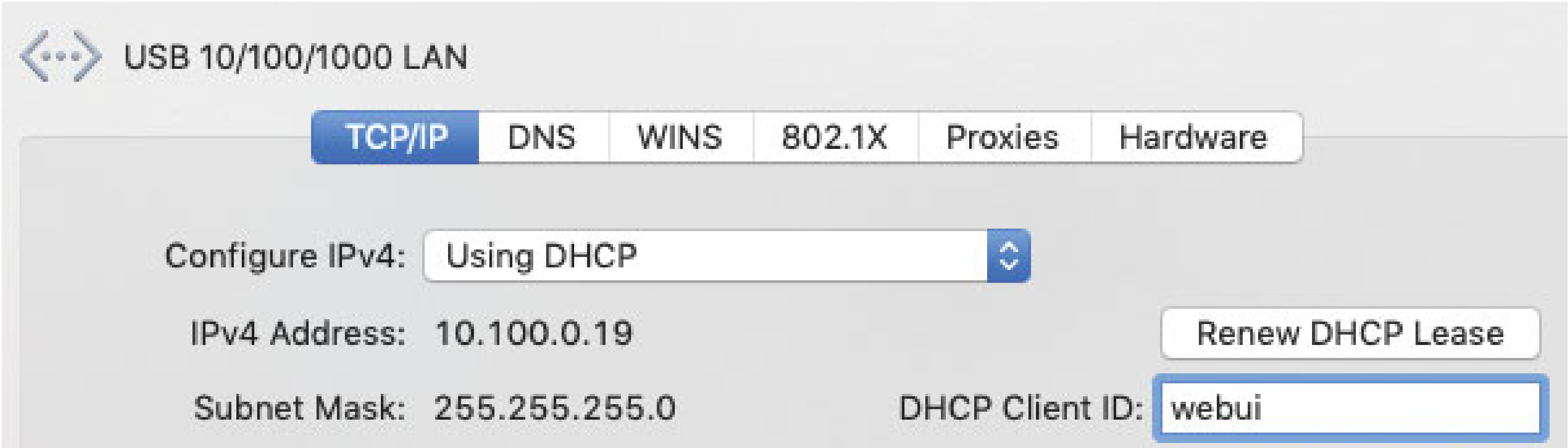


Figure 4-3 DHCP client ID setting on Mac

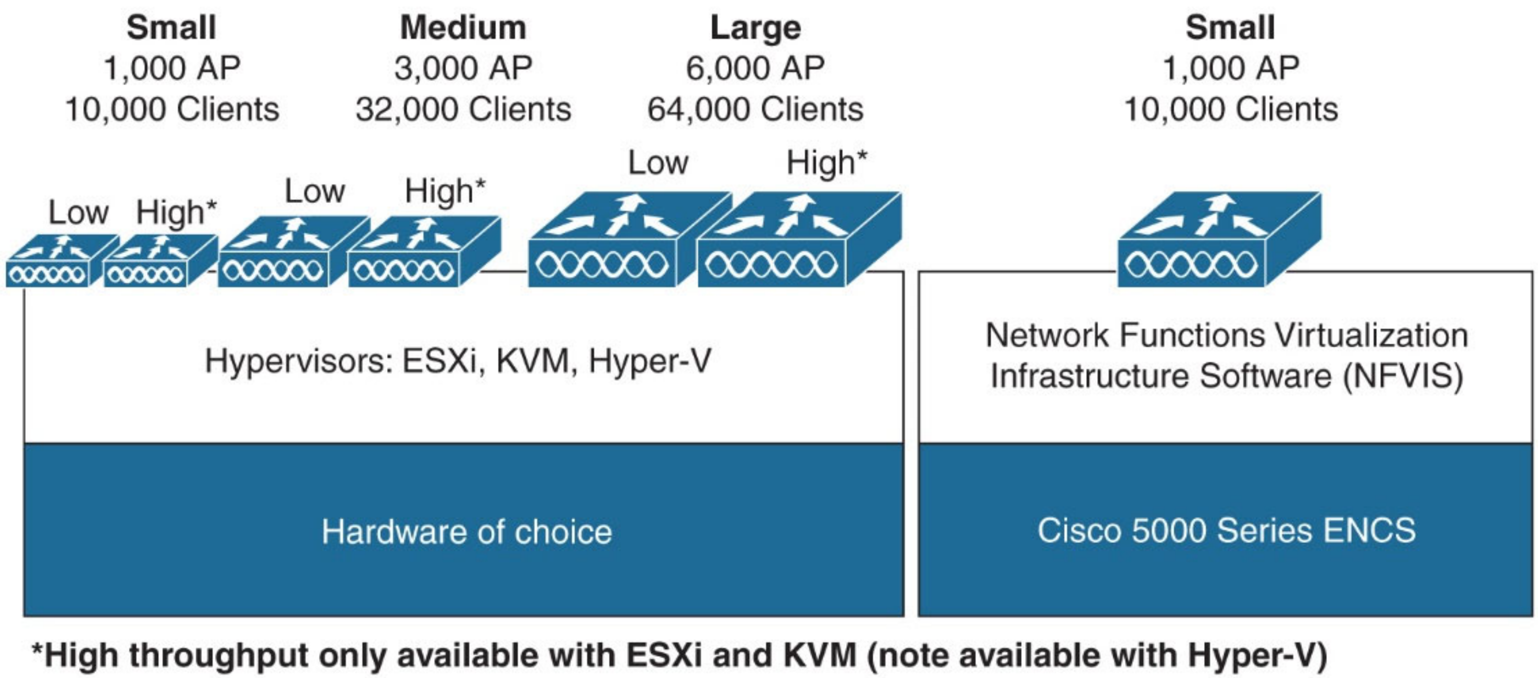


Figure 4-4 Cisco Catalyst 9800-CL for private cloud

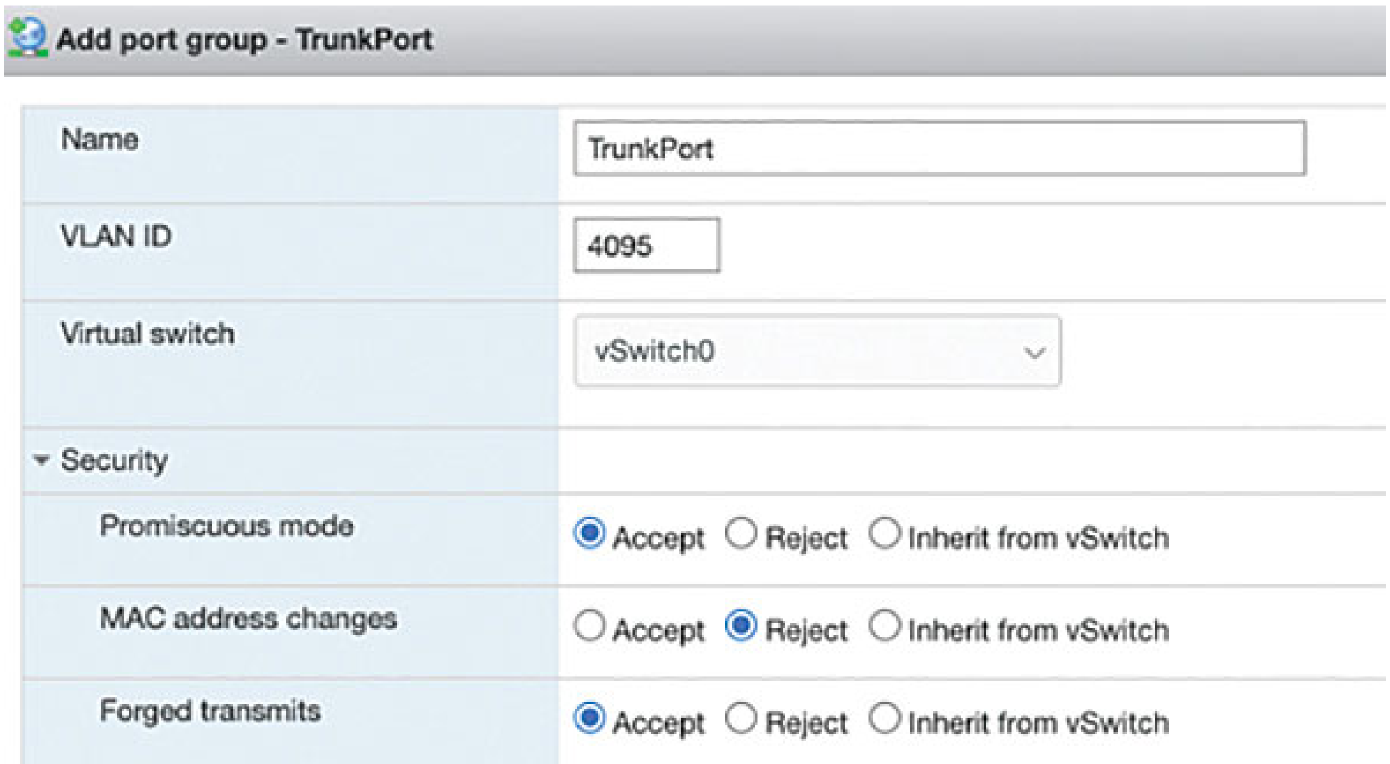
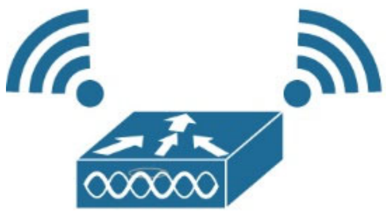


Figure 4-5 Security settings on VMware port group



Runs Cisco IOS XE C9800 Wireless Controller on Catalyst Access Points	Modern OS, scalable, open and programmable, supports telemetry
Supports Advanced Enterprise Feature Set	High Availability, aWIPS, Umbrella, SMU, ClearAir, FRA
Easy to deploy and manage	Use mobile app or WebUI to deploy, manage, and monitor, Cisco DNA Center
Investment protection	Migrate Access Points to controller for more than 100 Access Points

Figure 4-6 Cisco EWC-AP characteristics

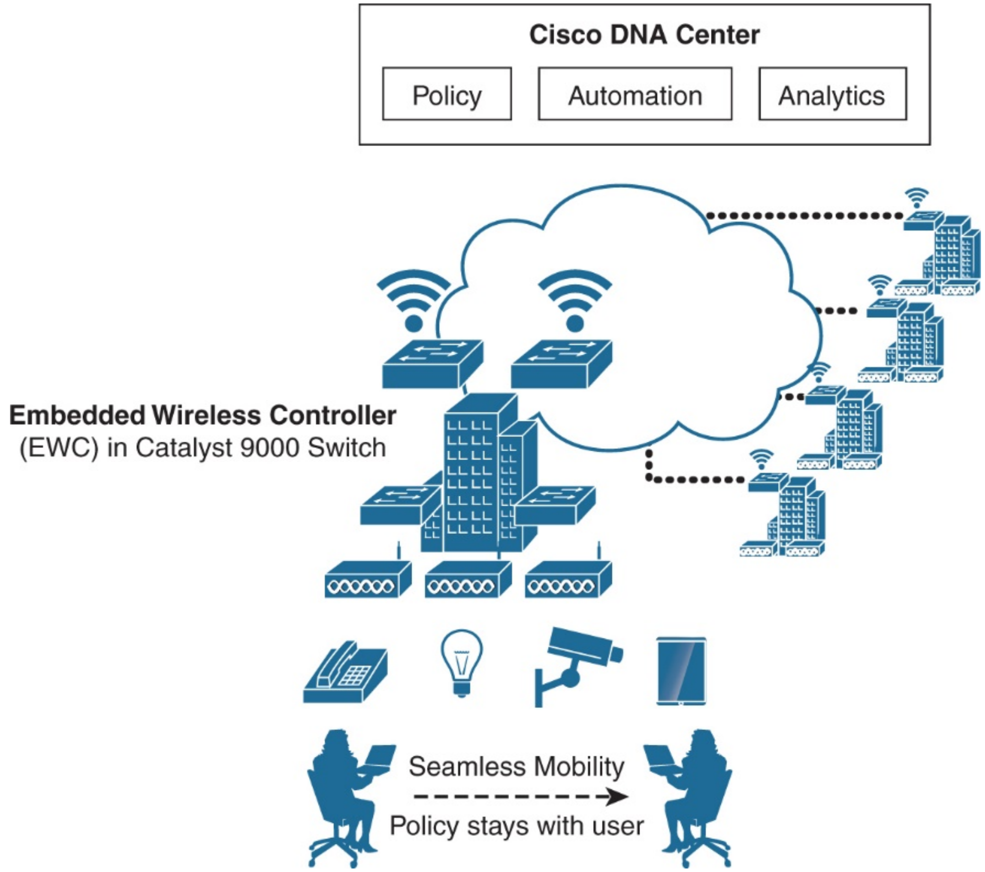


Figure 4-7 Cisco Embedded Wireless Controller (EWC) in Catalyst 9000 switches for SDA

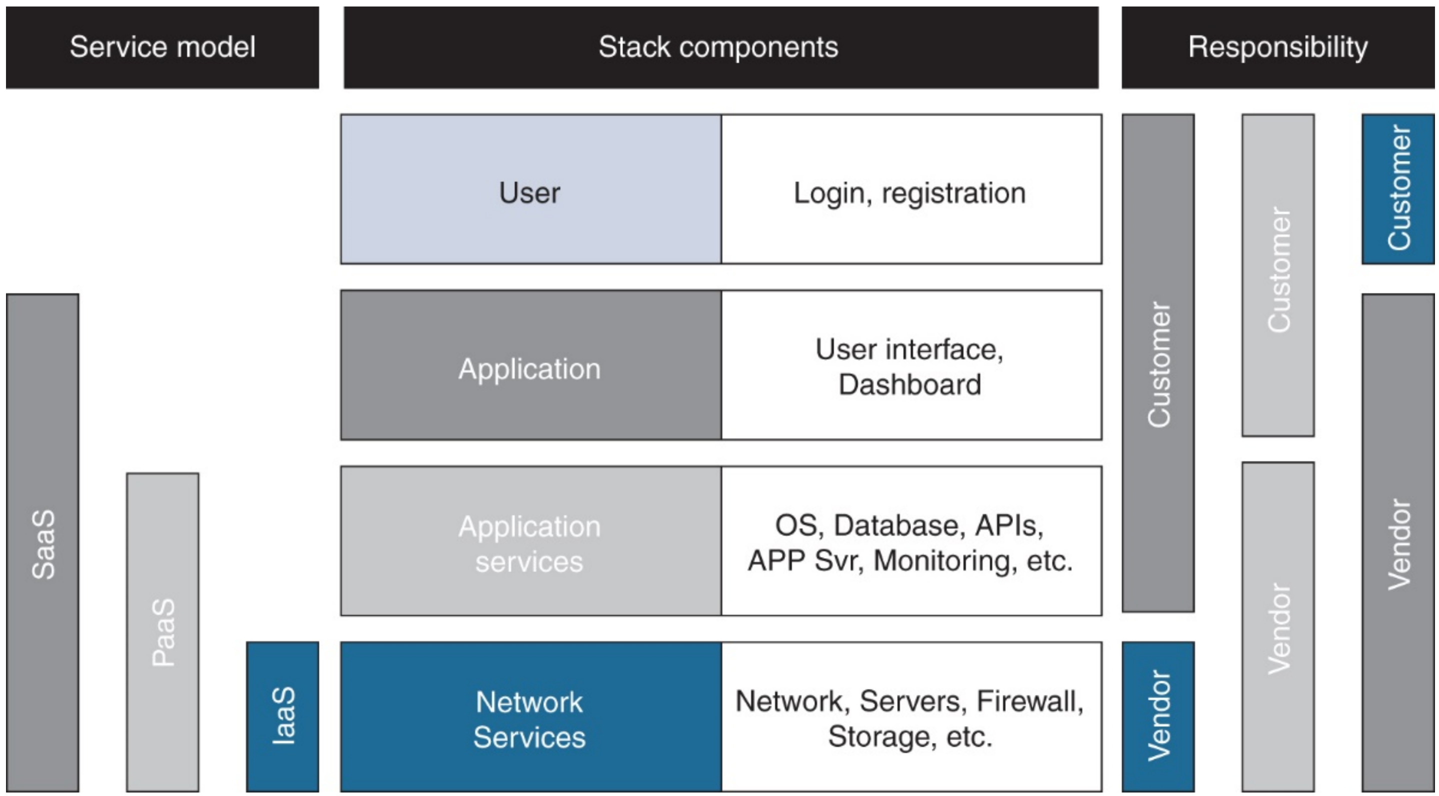


Figure 4-8 Public cloud service models

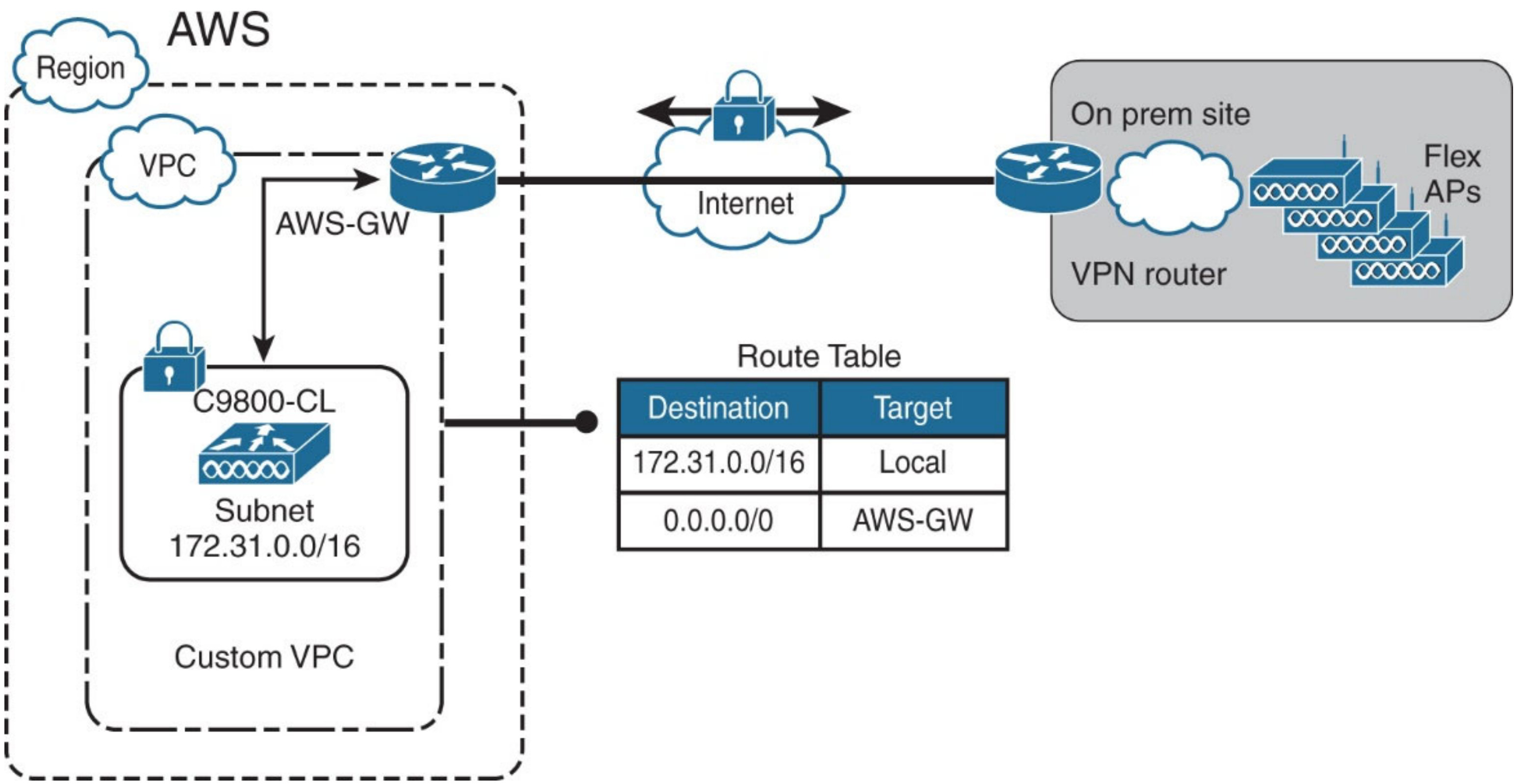


Figure 4-9 Managed VPN deployment mode on the public cloud

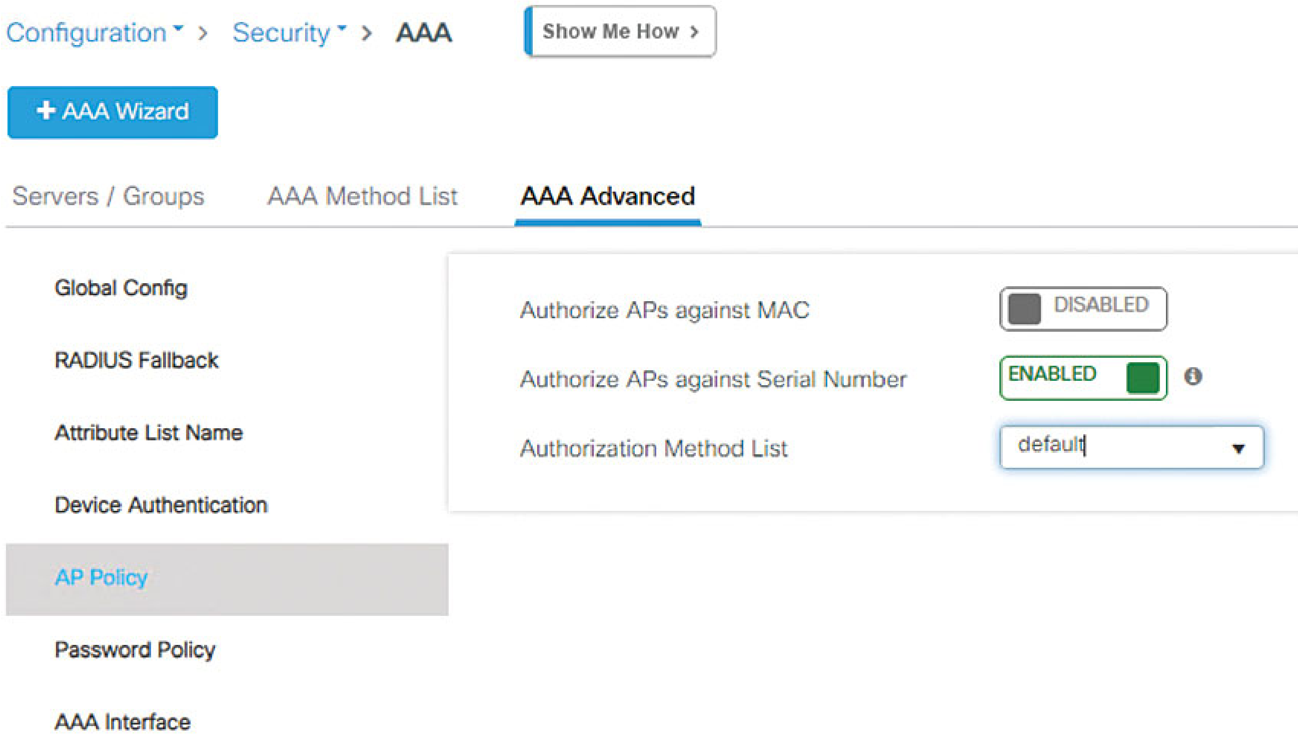


Figure 4-10 Authorize APs against serial number

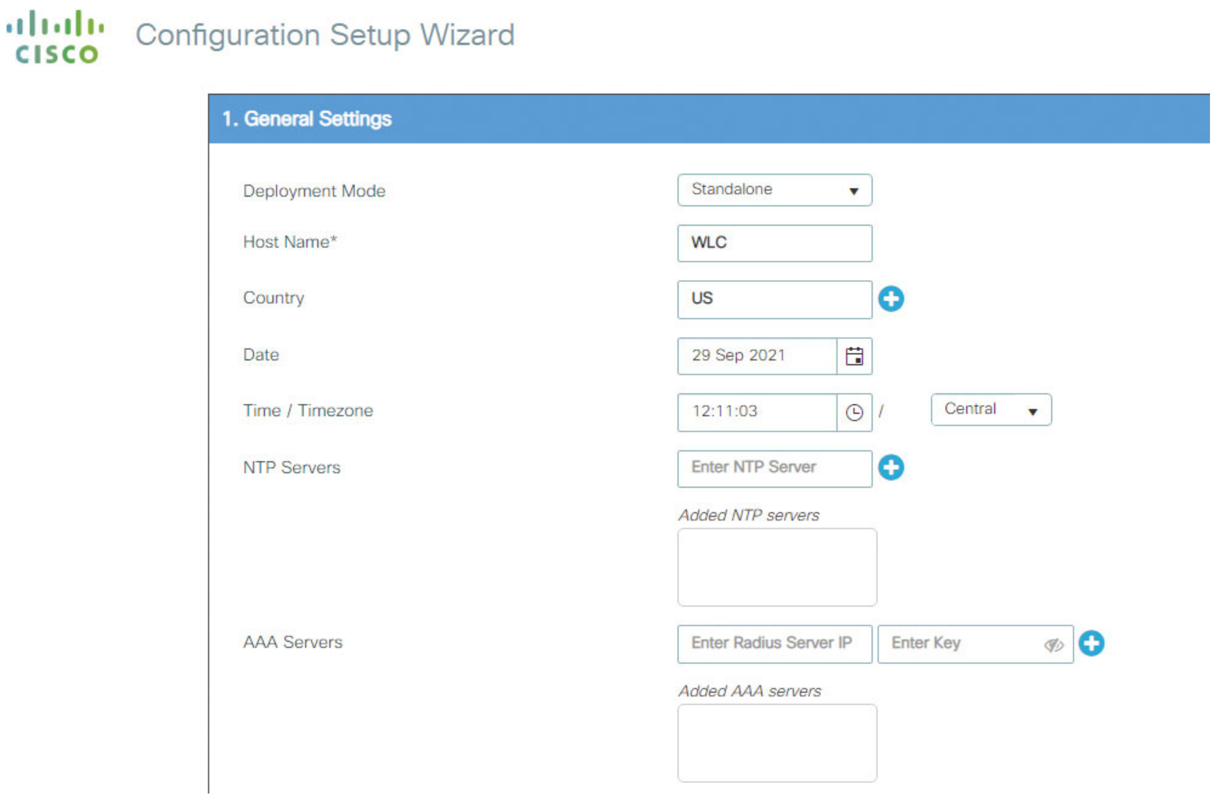


Figure 4-11 Day 0 Configuration Setup Wizard

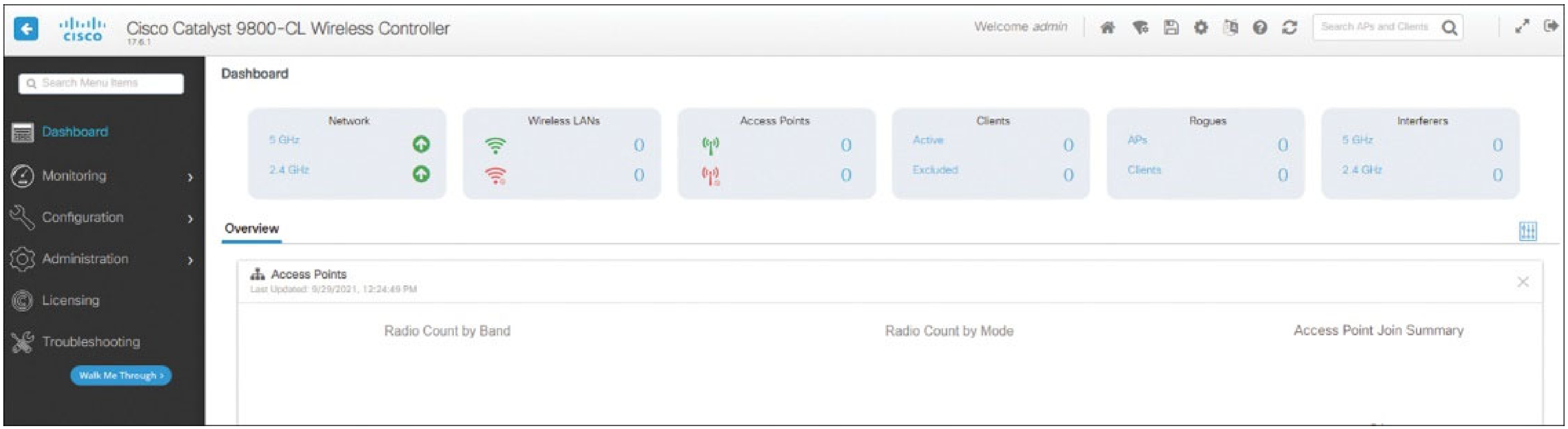


Figure 4-12 Catalyst 9800 wireless LAN controller main dashboard

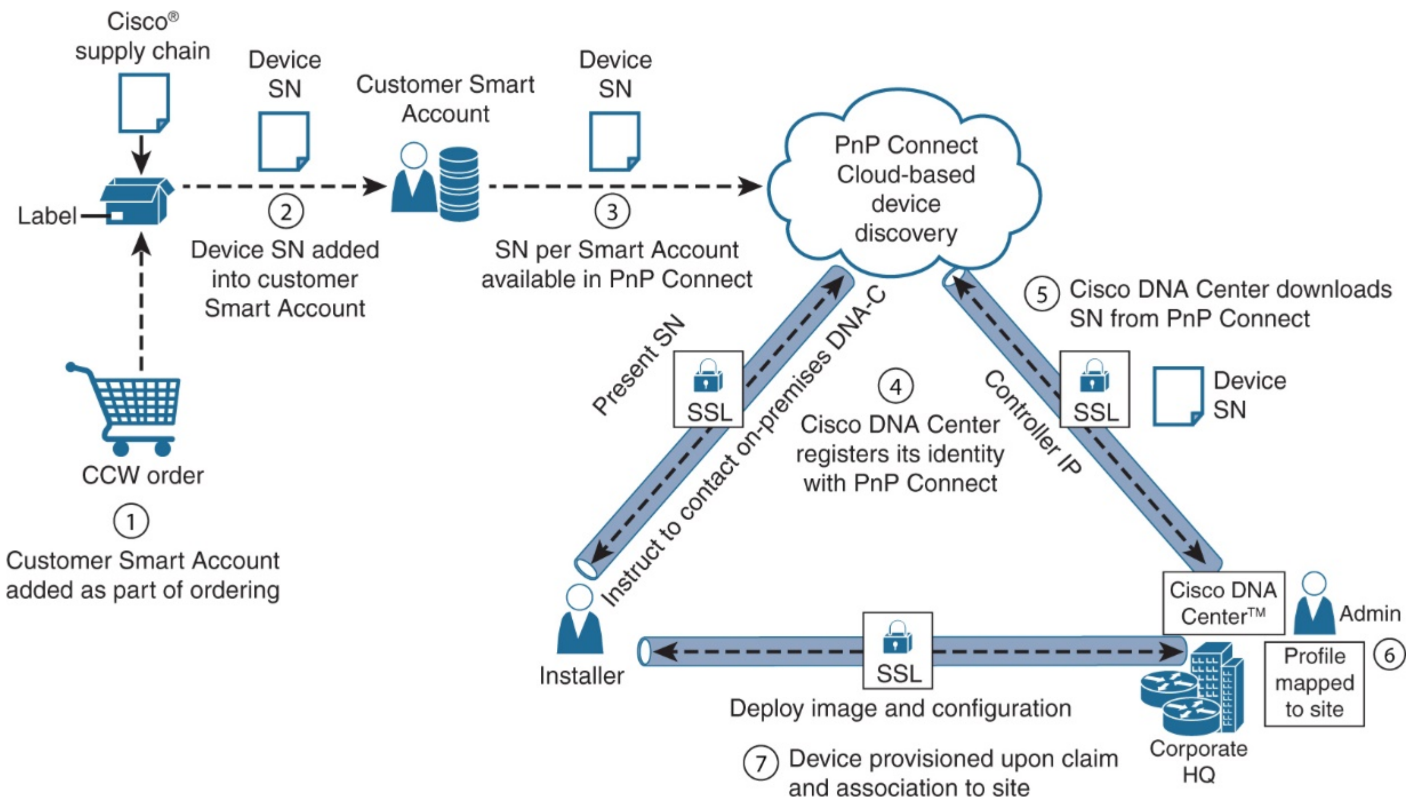


Figure 4-13 AP end-to-end onboarding process with PnP

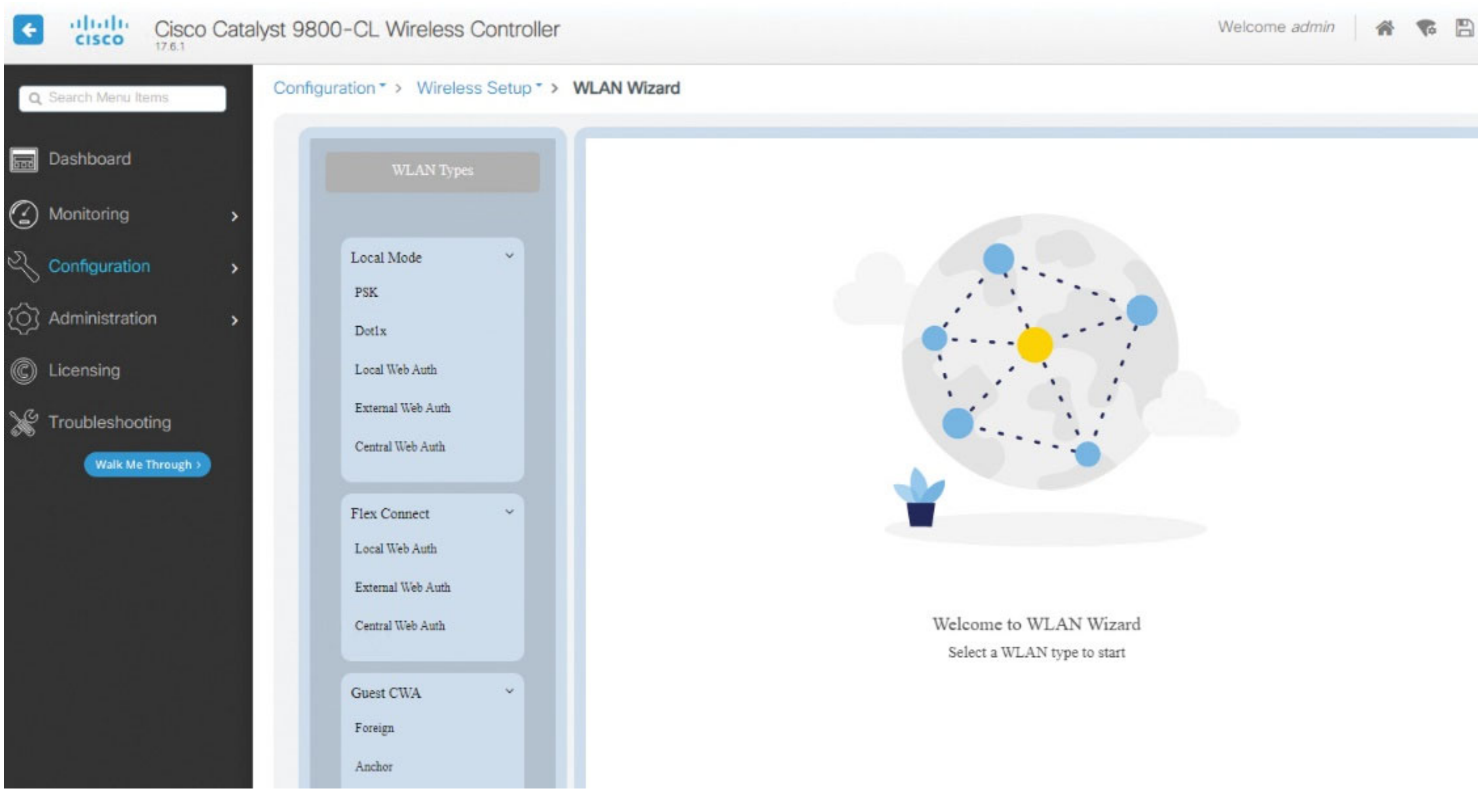


Figure 4-14 WLAN Wizard

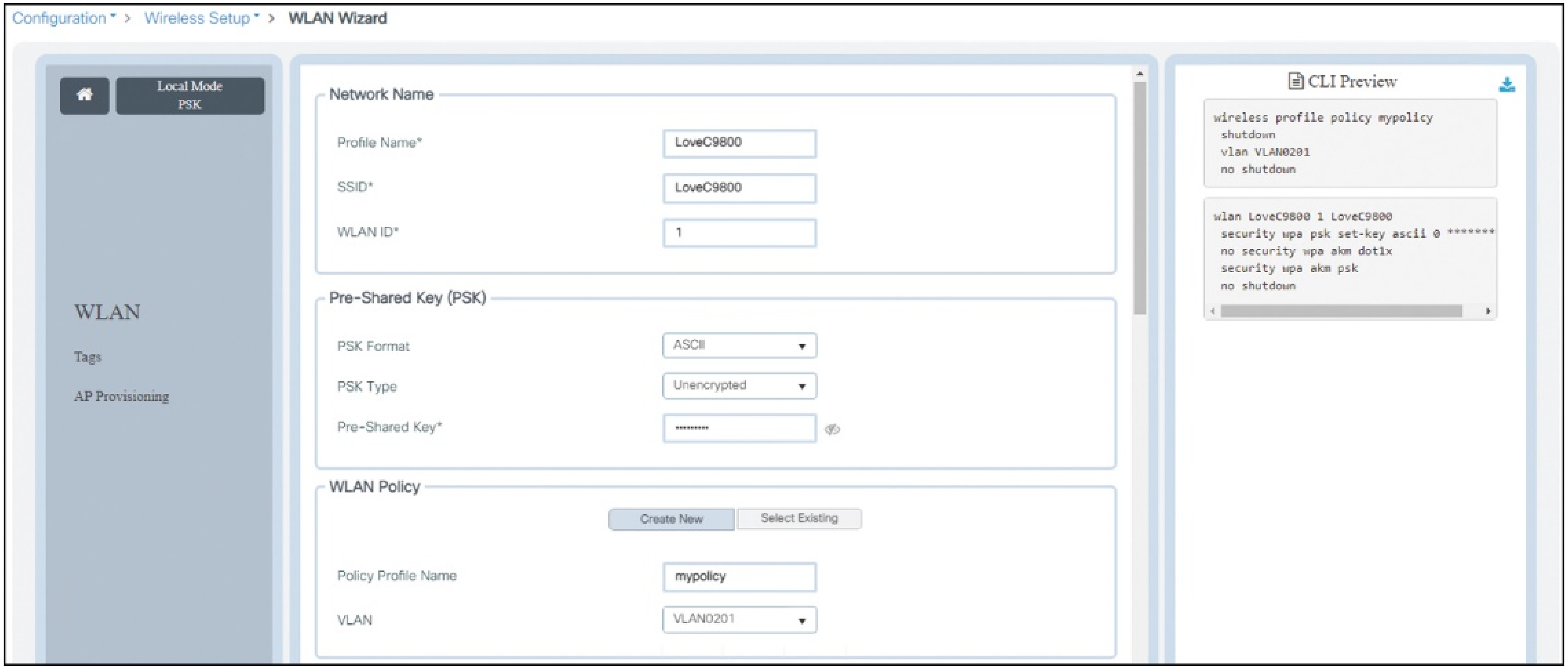


Figure 4-15 WLAN Wizard, CLI preview

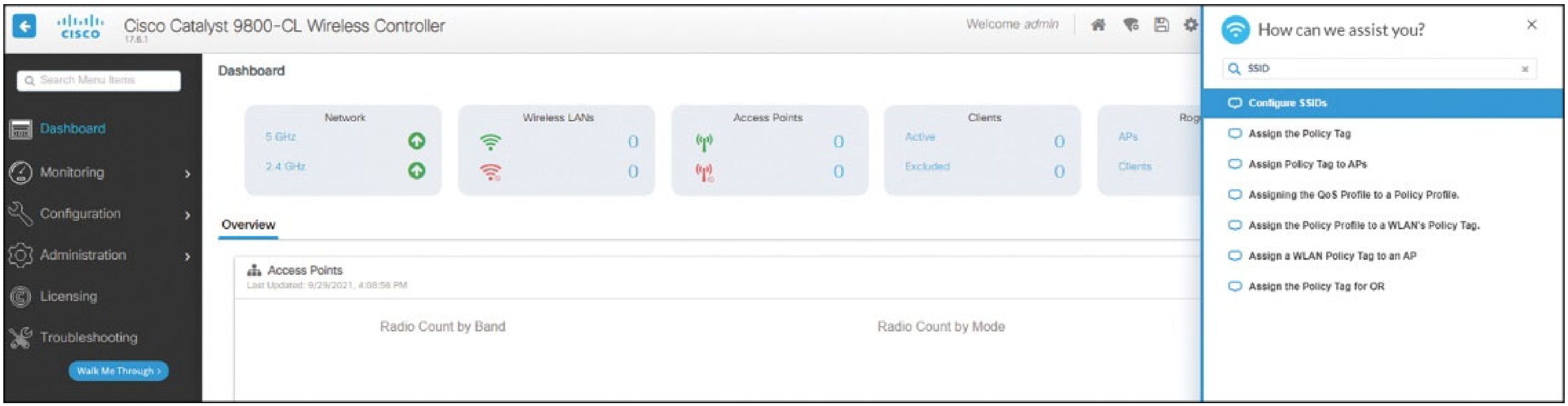


Figure 4-16 Walk Me configuration tool

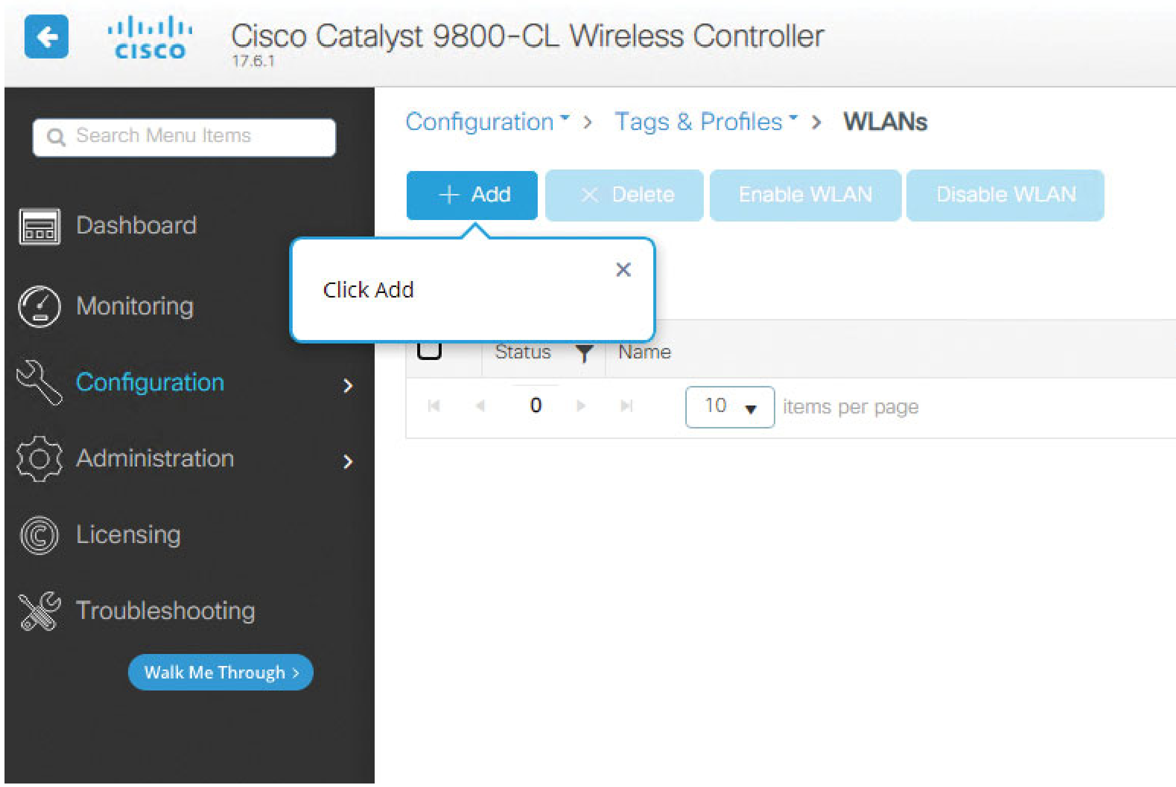


Figure 4-17 WLAN creation in Walk Me

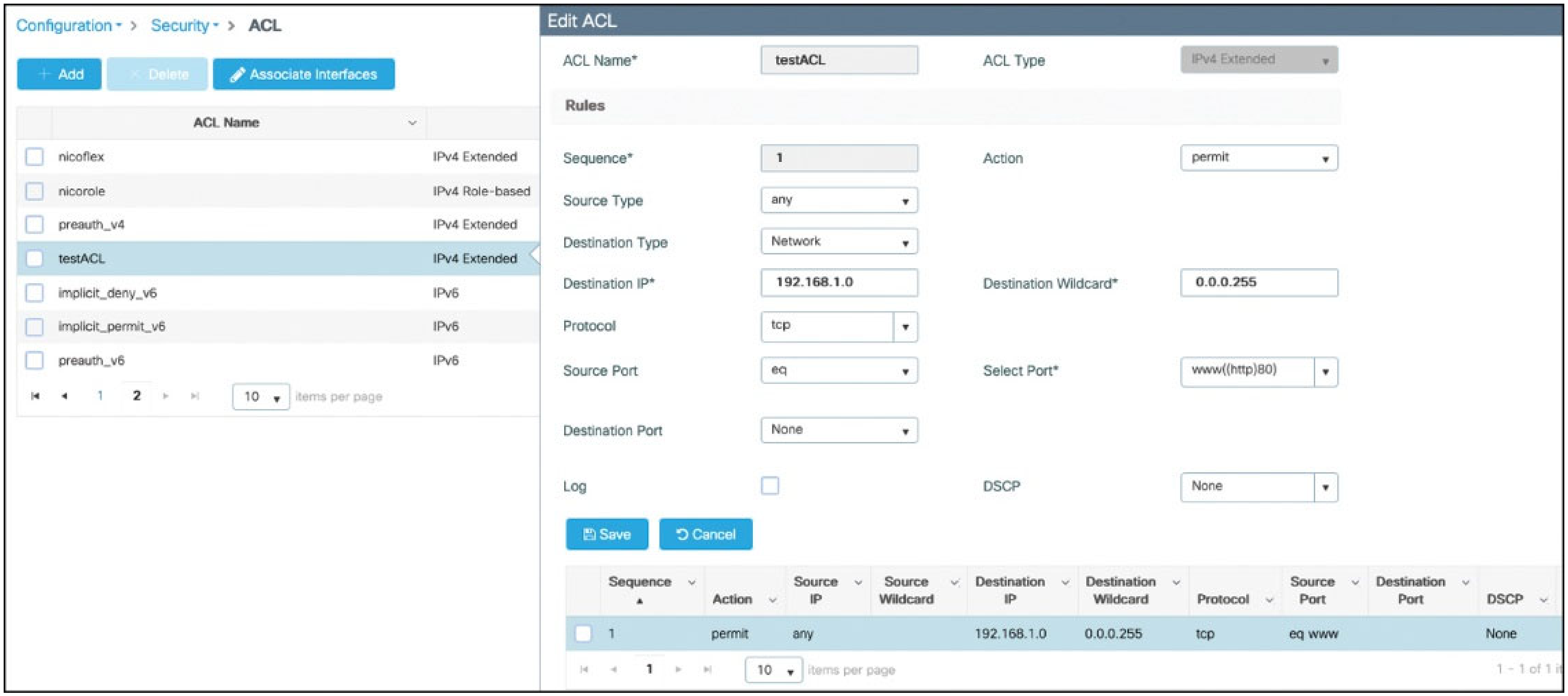


Figure 5-1 Creating an ACL on the Catalyst 9800

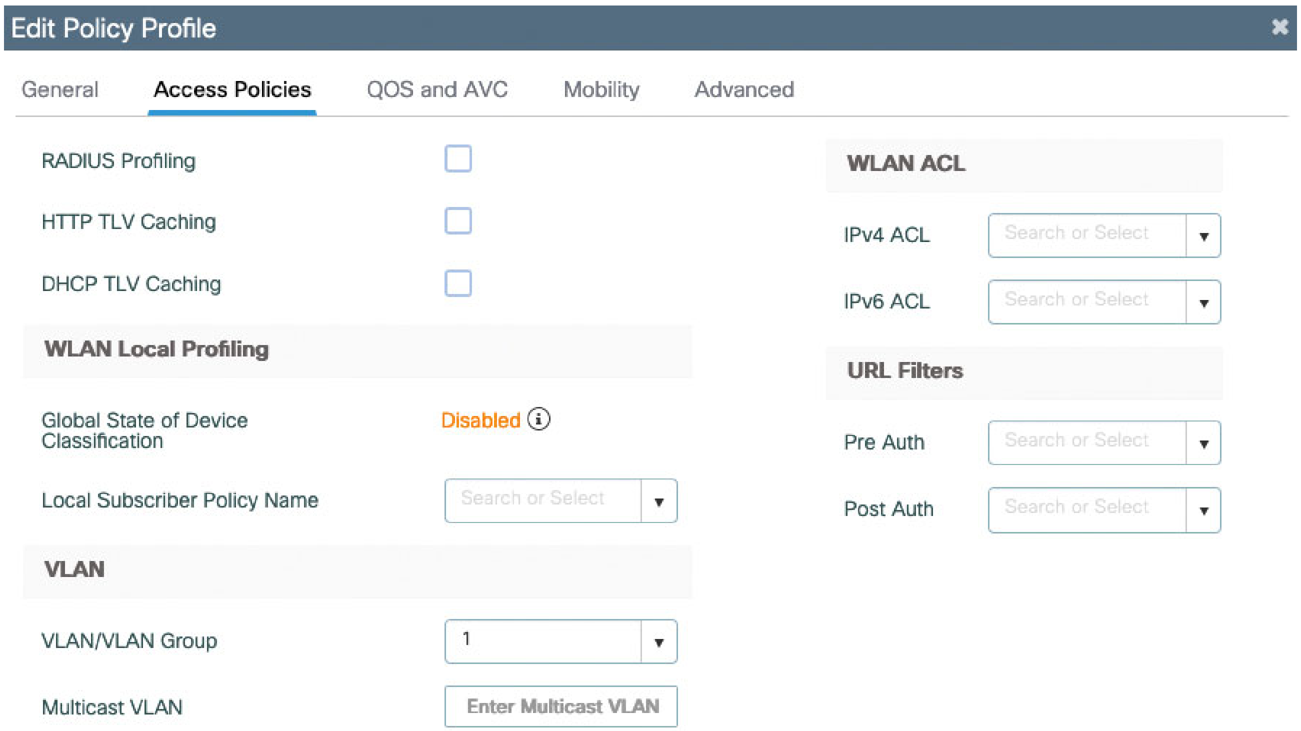


Figure 5-2 Access policies of the policy profile on the C9800

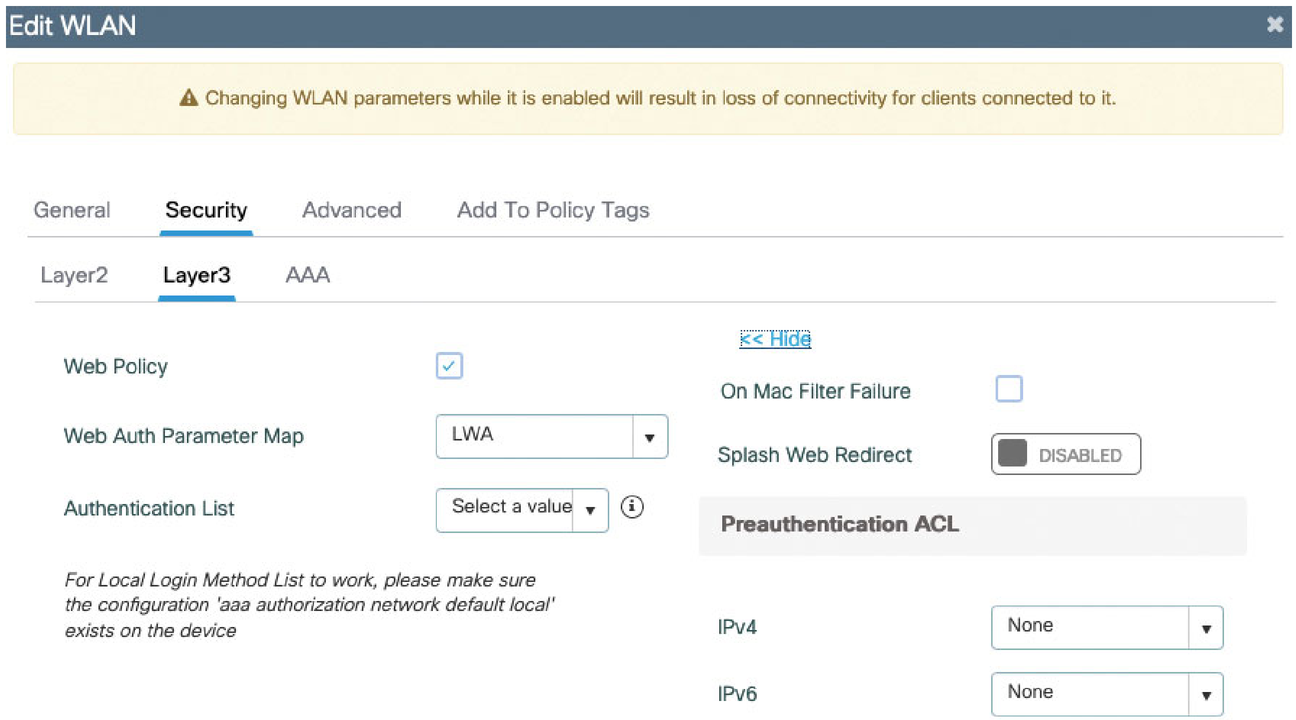


Figure 5-3 Layer3 WLAN security settings where the preauthentication ACL can be configured

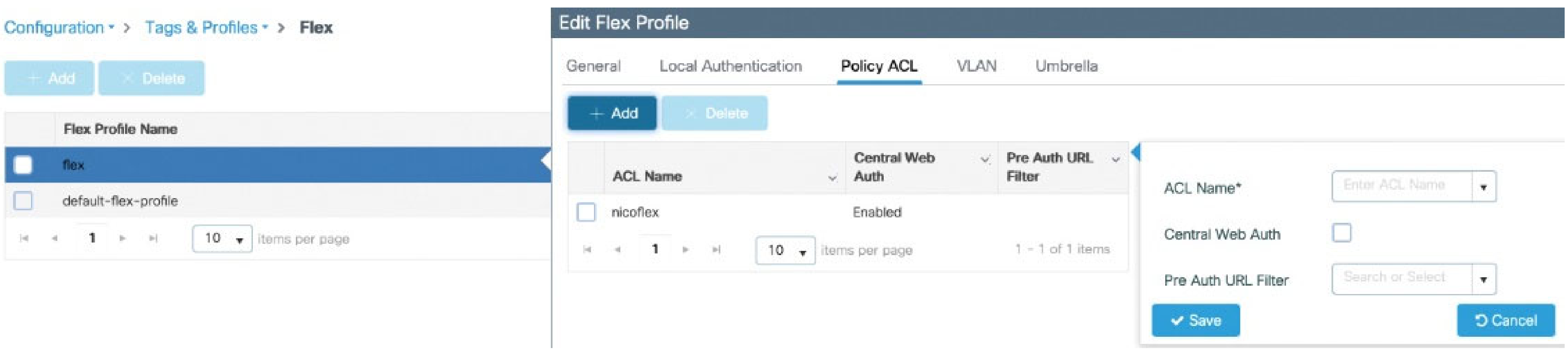


Figure 5-4 The Flex profile Policy ACL tab allows you to download ACLs to the APs

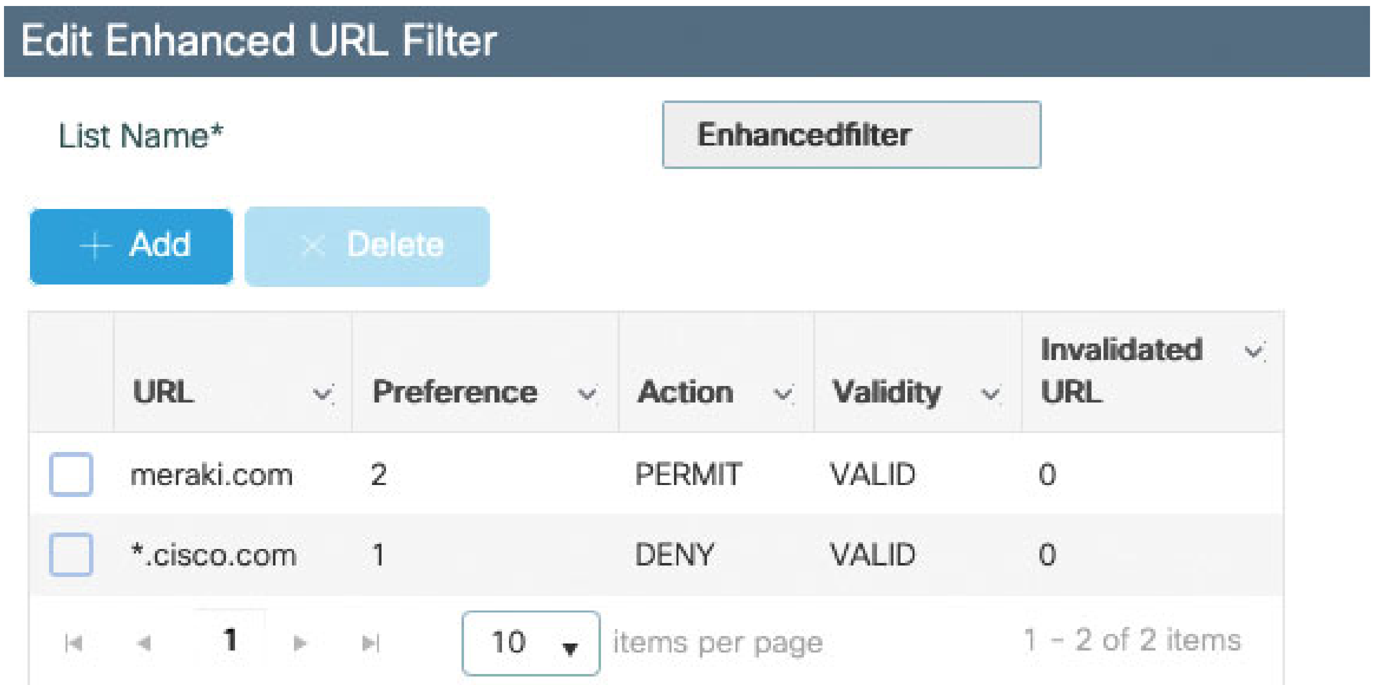


Figure 5-5 URL filter configuration

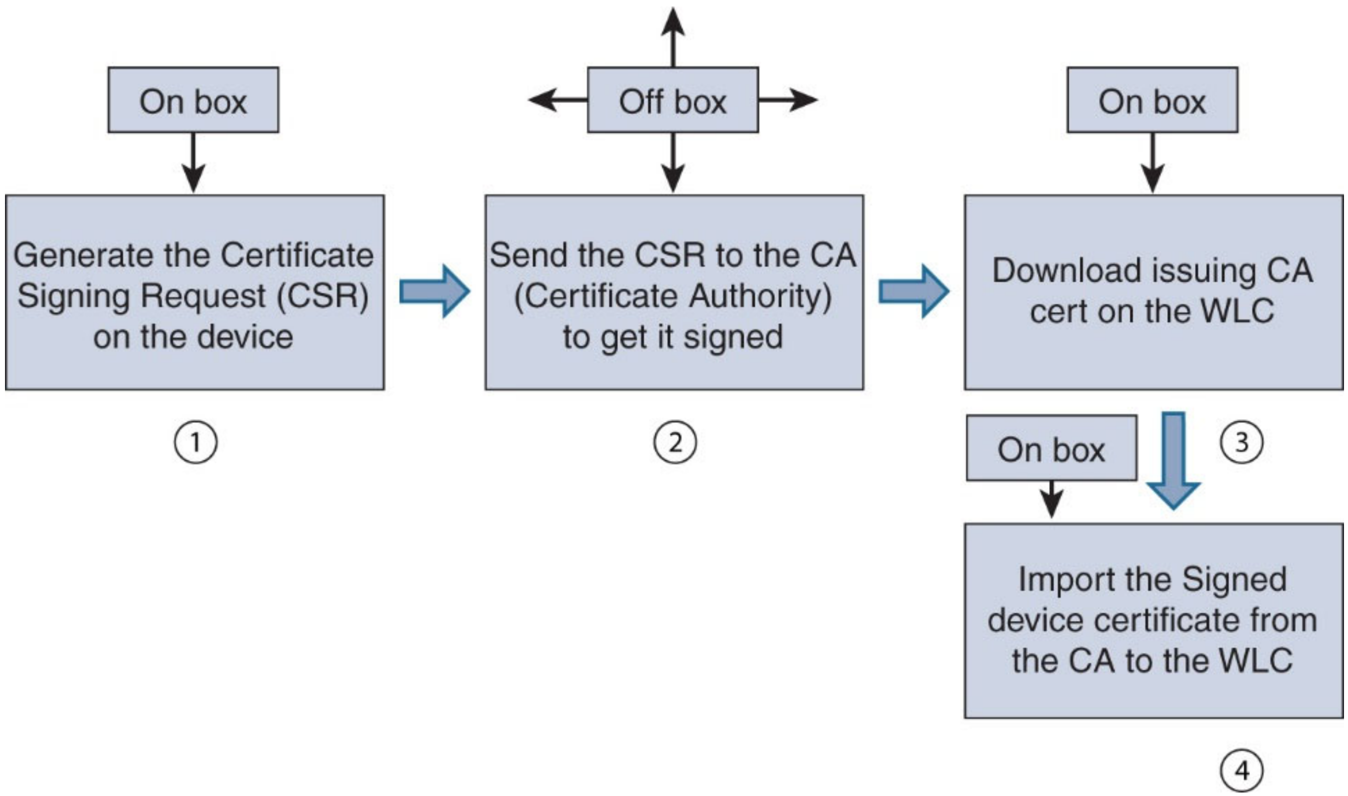


Figure 5-6 Adding a certificate by generating the question on the controller

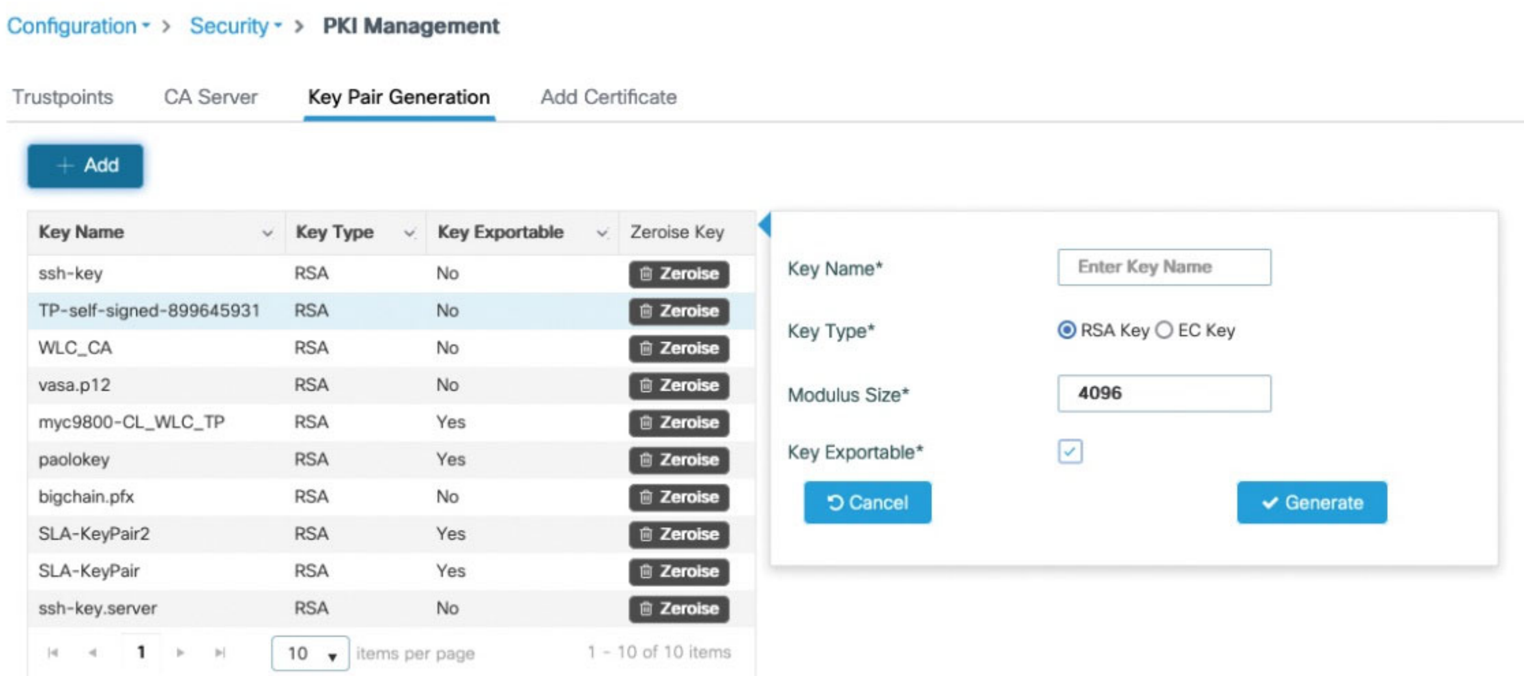


Figure 5-7 Creating a key pair for use with a certificate

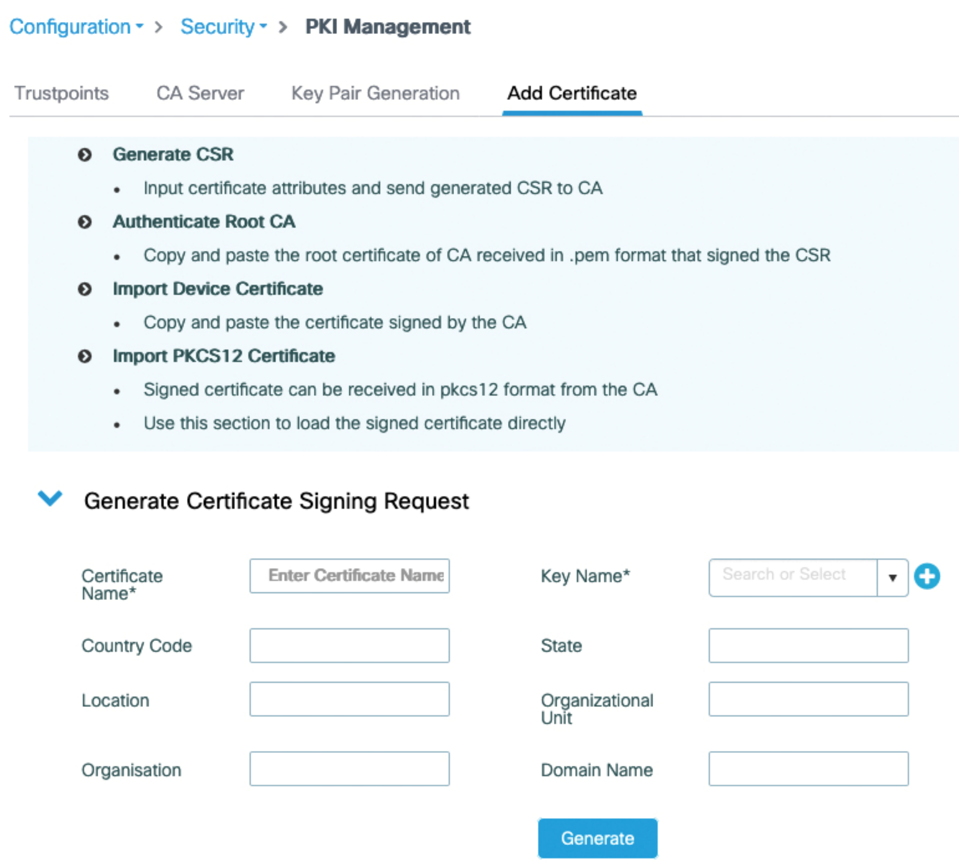


Figure 5-8 Generate a certificate signing request from the WebUI

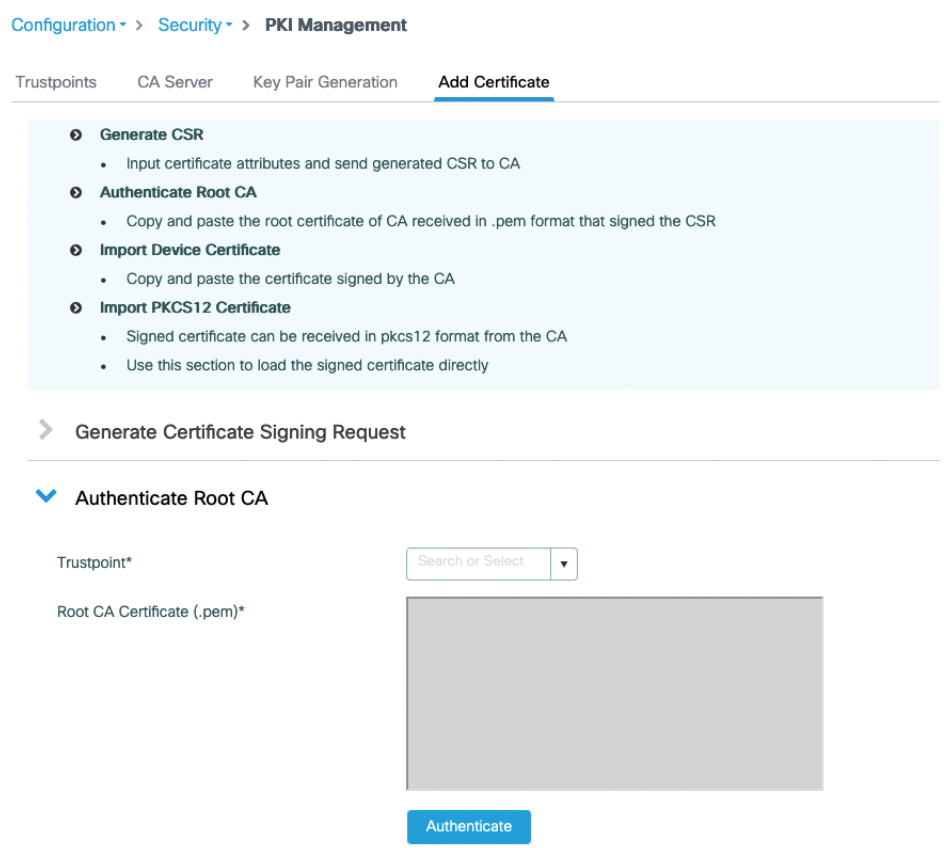


Figure 5-9 Authenticating the CA that issued your device certificate

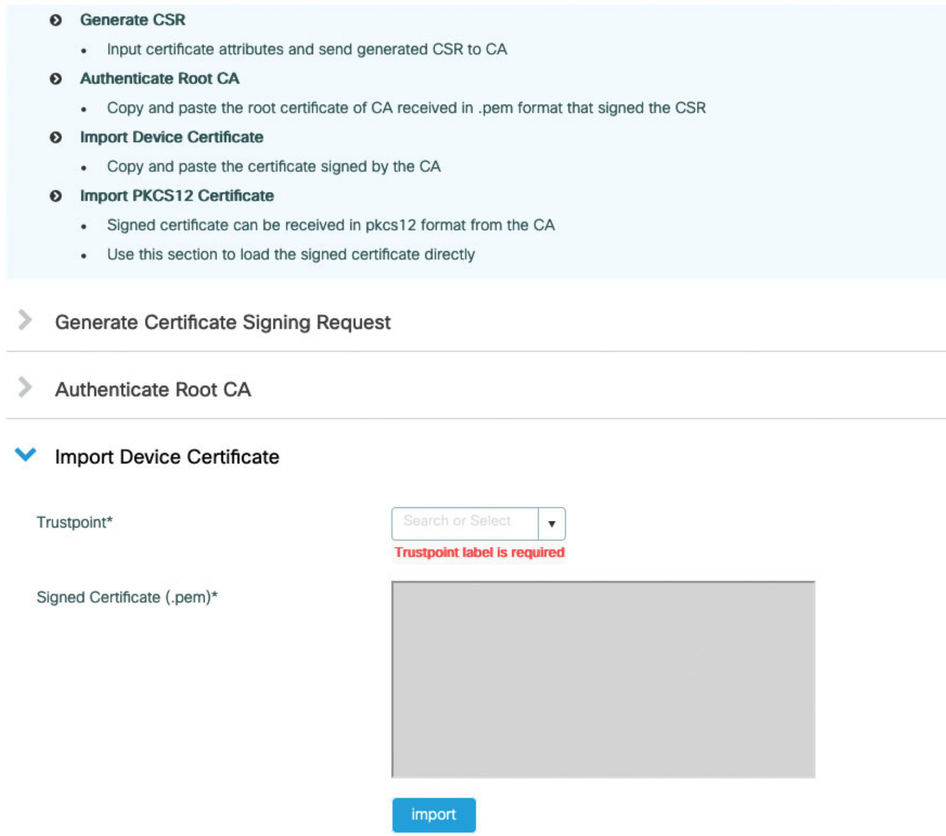


Figure 5-10 Adding the device signed certificate received from the CA

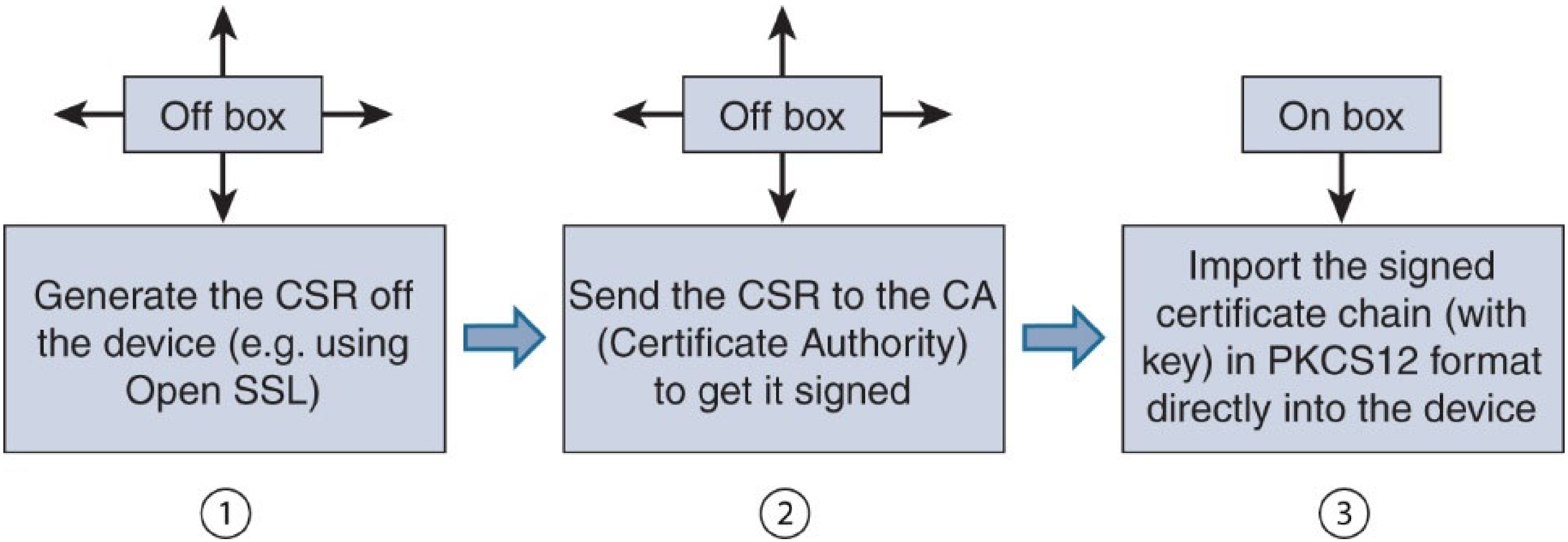


Figure 5-11 Adding a certificate when the private key was generated outside of the 9800

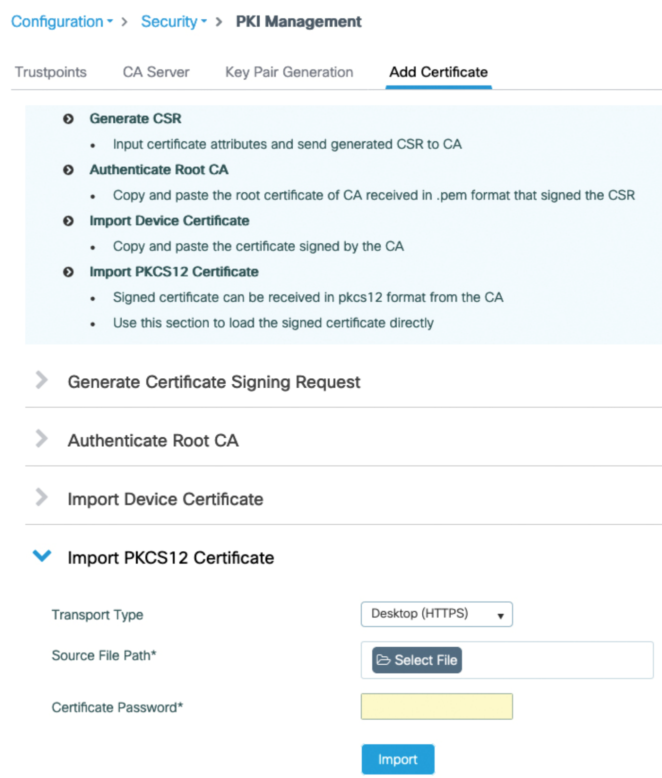


Figure 5-12 Adding a certificate chain in PKCS12 format to the WLC

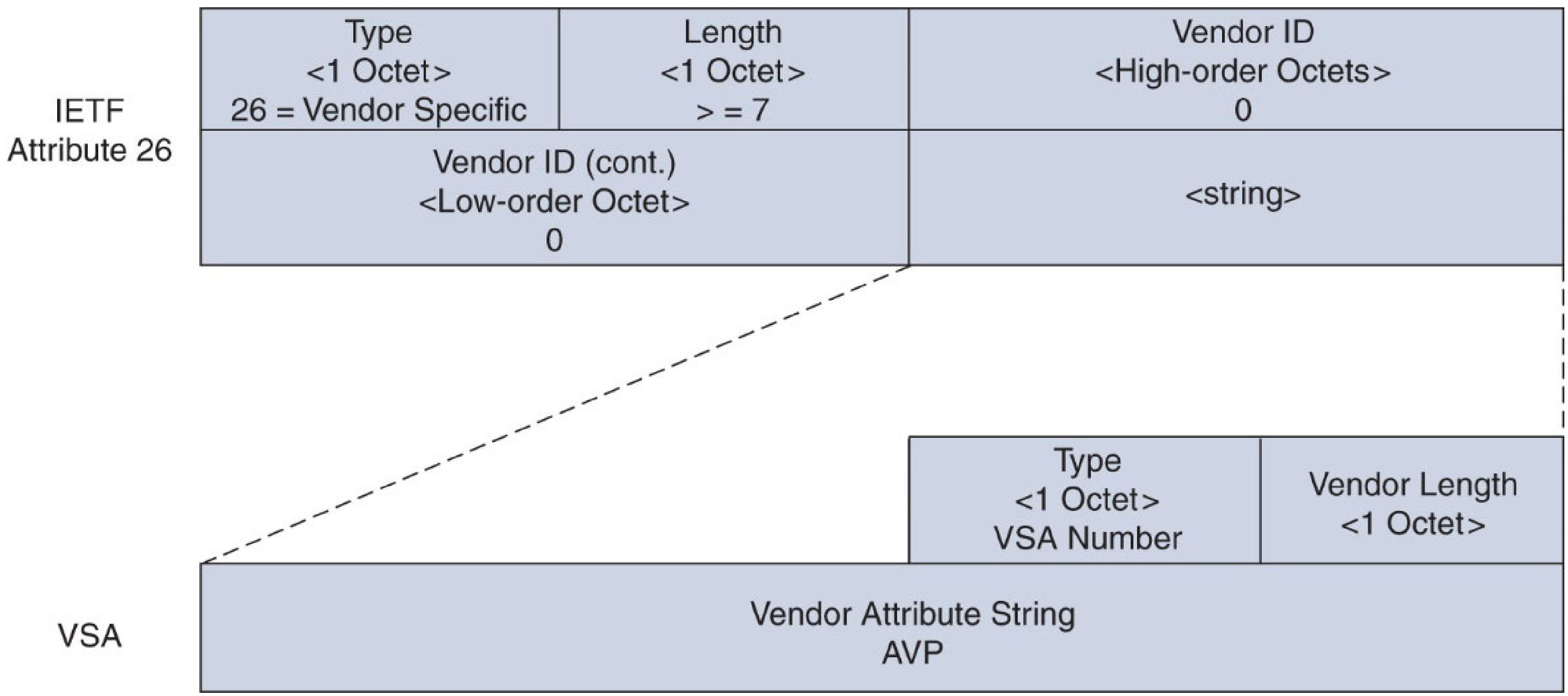


Figure 5-13 RADIUS AV-pair format

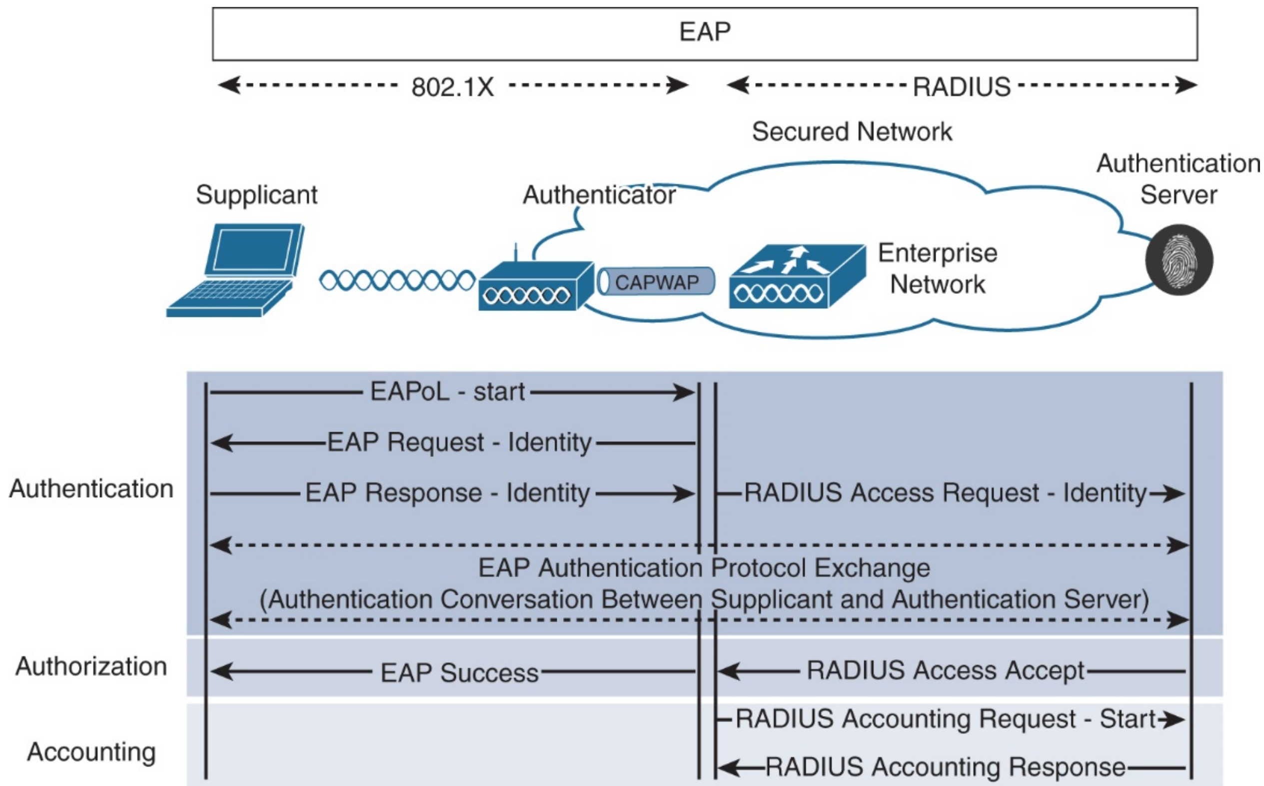


Figure 5-14 Workflow of an EAP authentication between a supplicant and authentication server

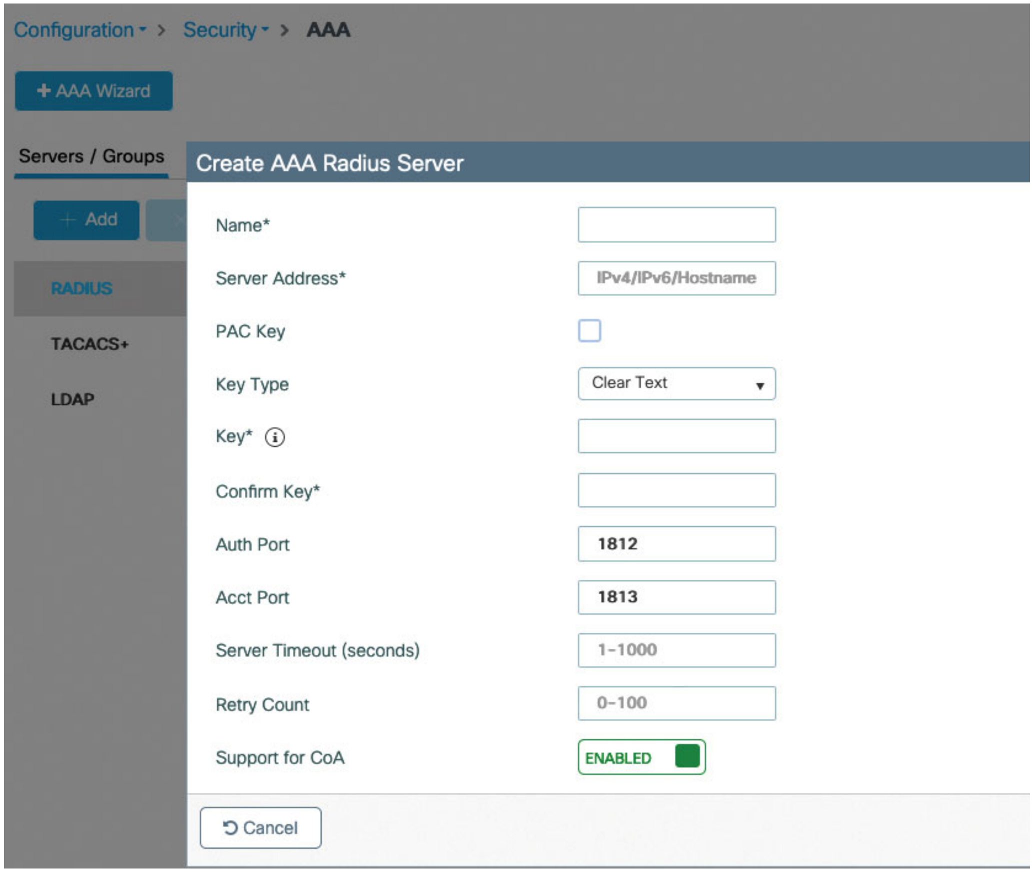


Figure 5-15 RADIUS server creation on the C9800 WebUI

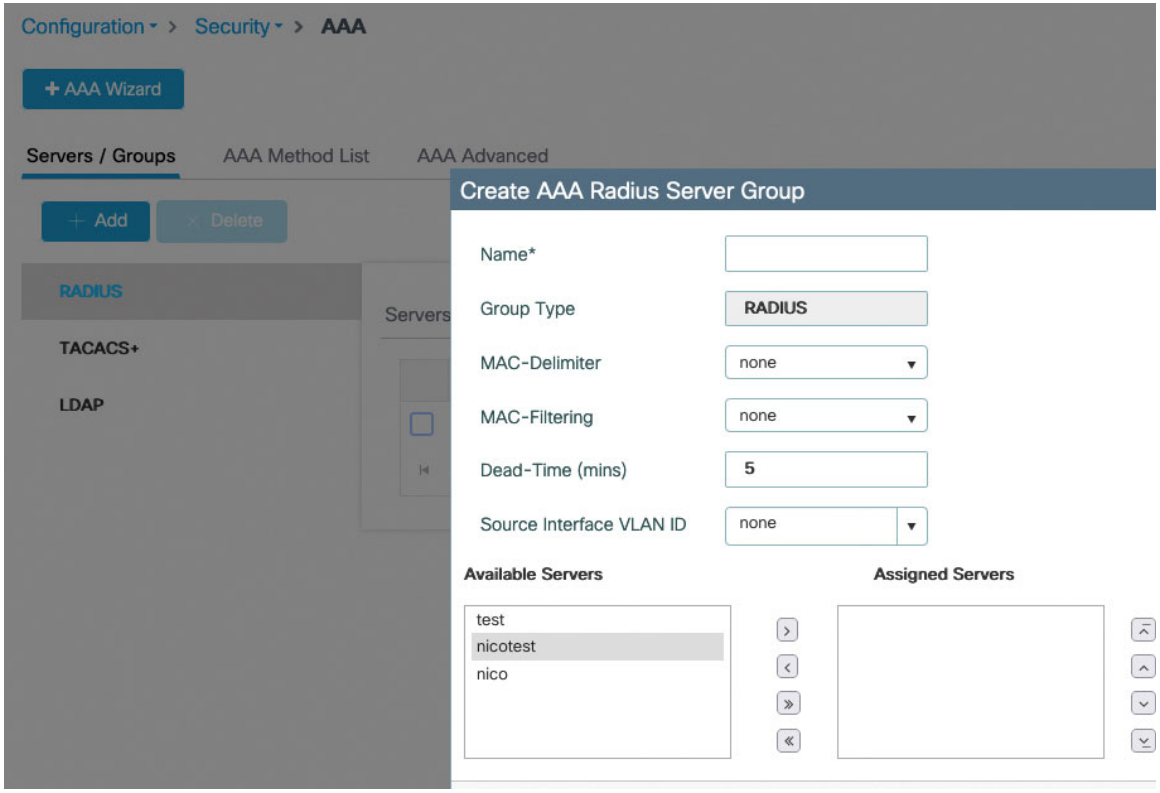


Figure 5-16 RADIUS server group configuration

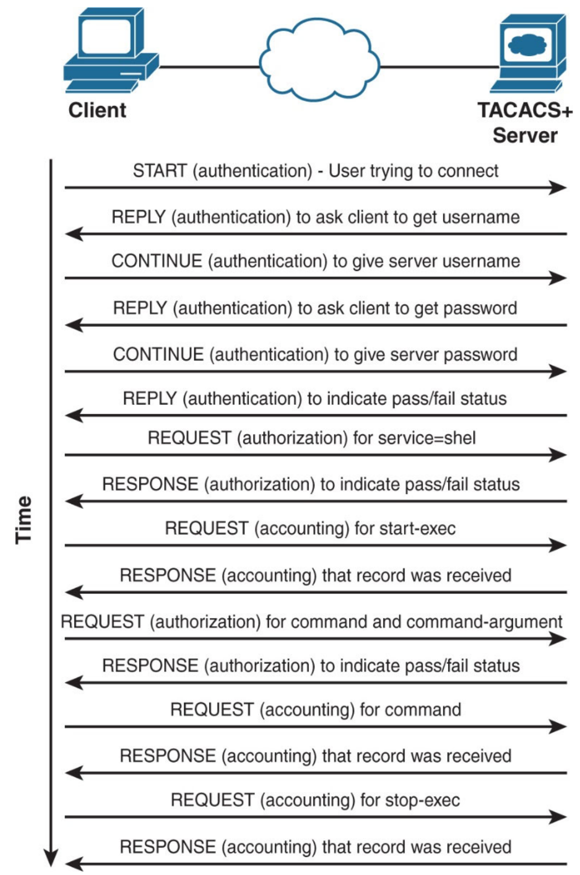


Figure 5-17 TACACS workflow

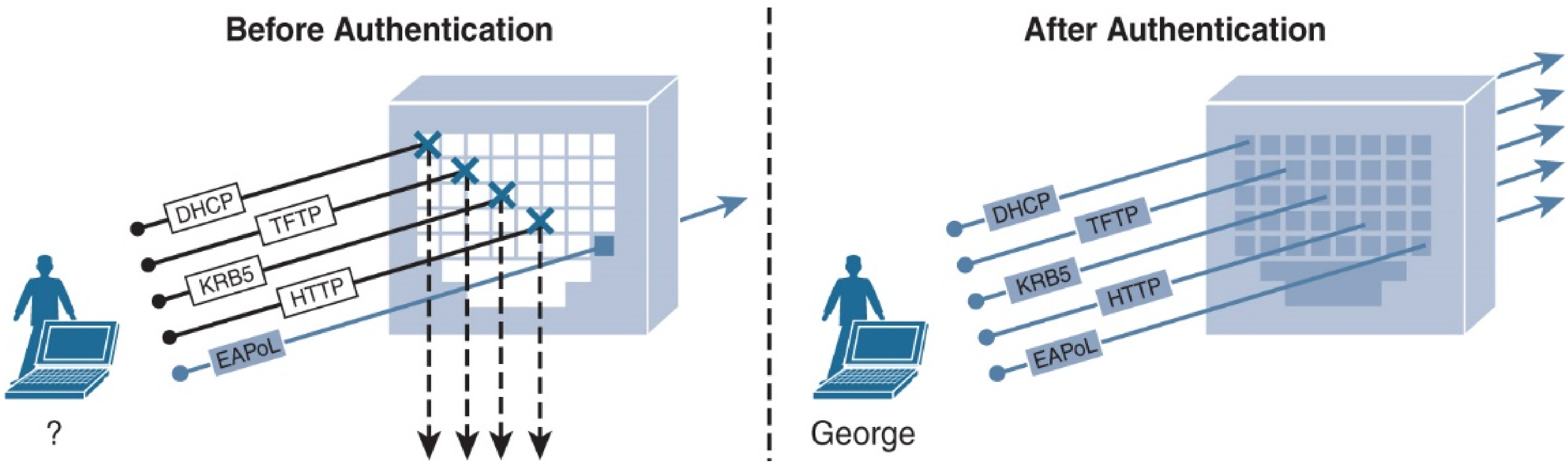


Figure 5-18 802.1X port-based authentication system

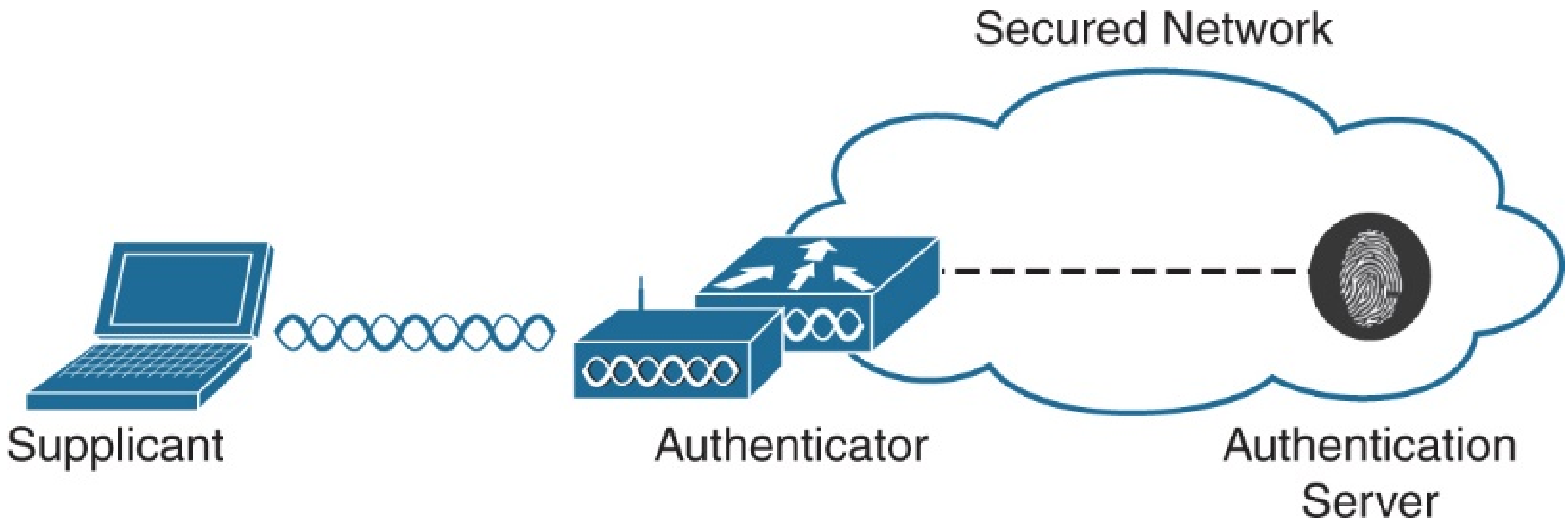


Figure 5-19 802.1X components

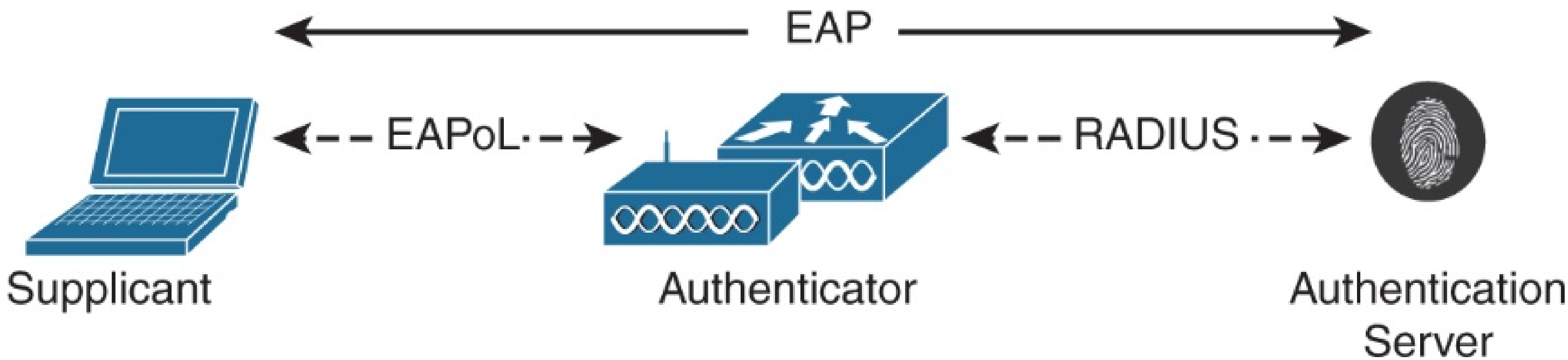


Figure 5-20 EAP authentication workflow

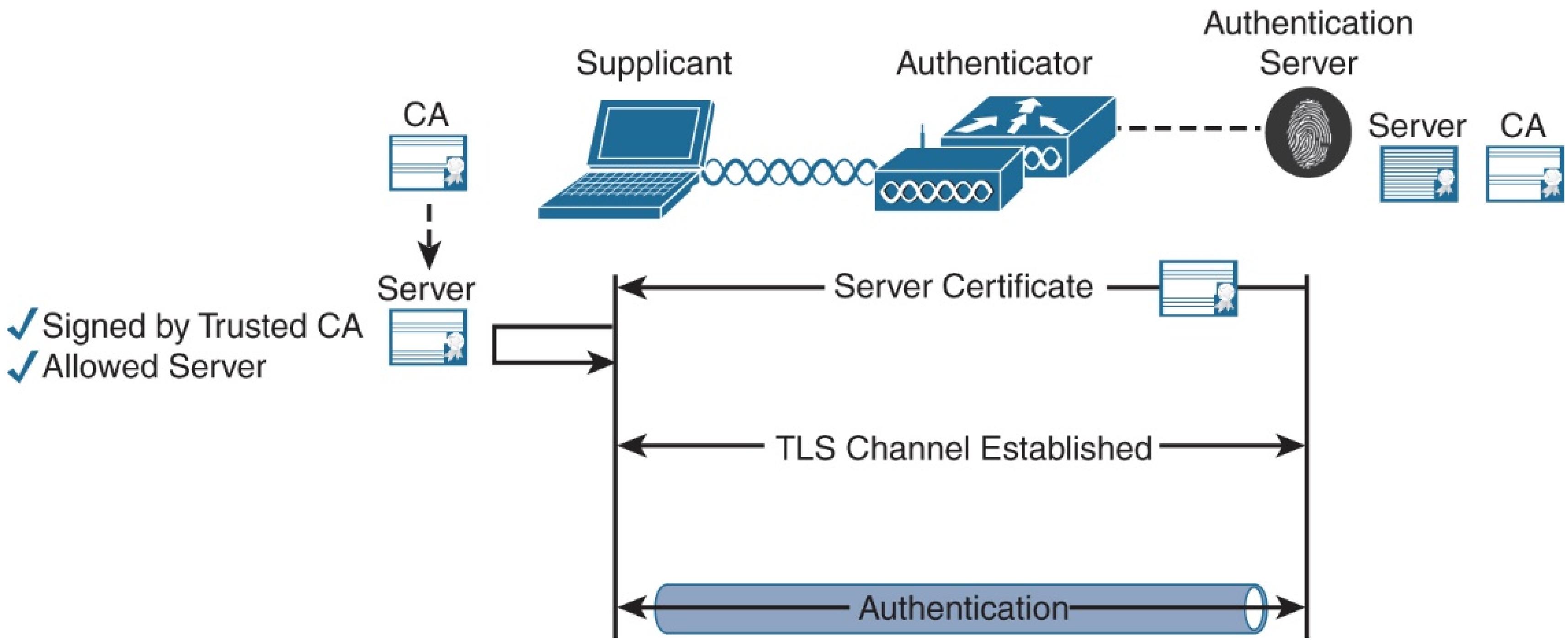


Figure 5-21 PEAP authentication high-level workflow

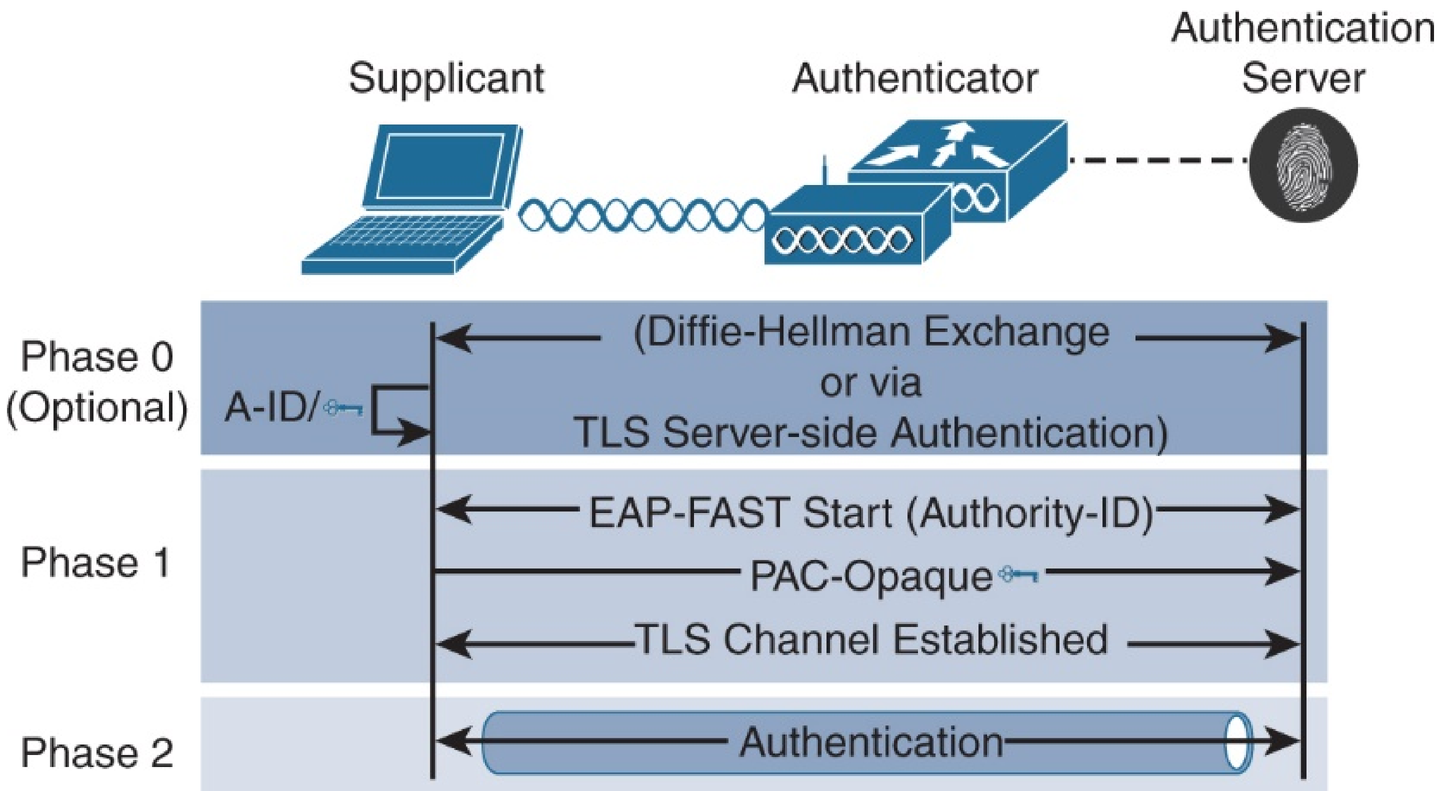


Figure 5-22 EAP-FAST high-level overview

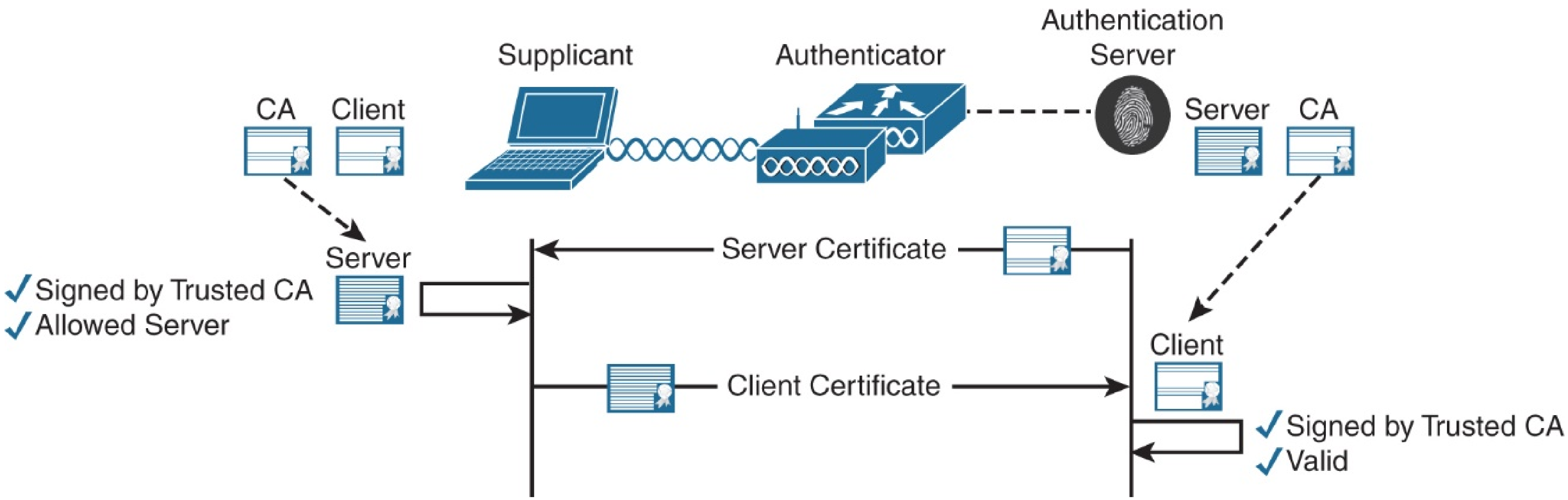


Figure 5-23 EAP-TLS high-level overview

From *Understanding and Troubleshooting Cisco Catalyst 9800 Series Wireless Controller by* Simone Arena, Francisco Sedano Crippa, Nicolas Darchis and Sudha Katgeri(9780137492329) Cisco Press

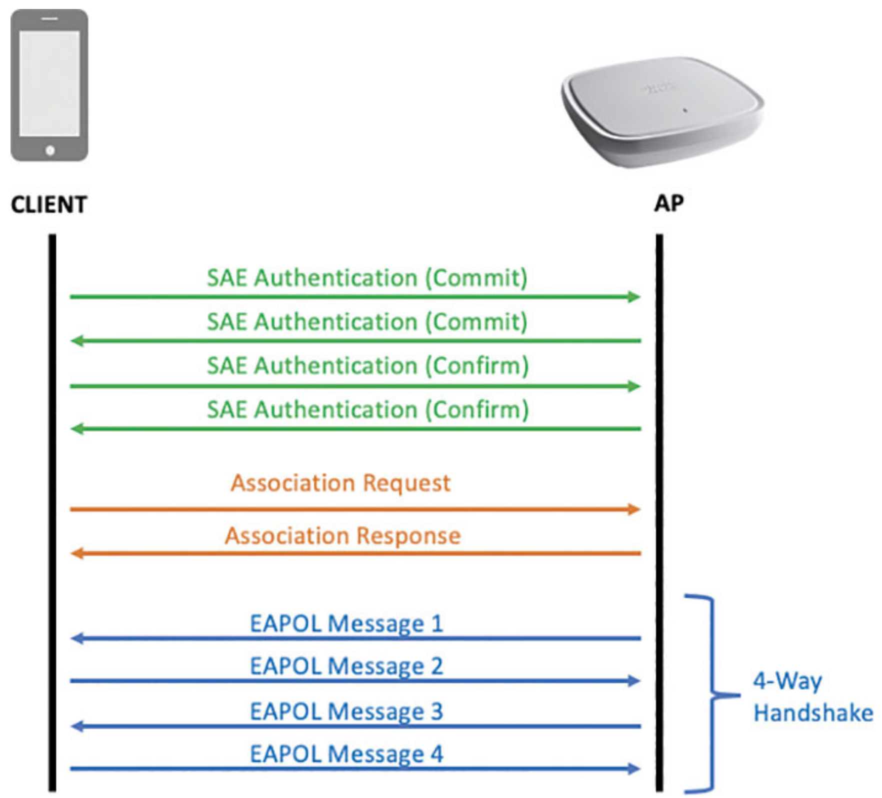


Figure 5-24 WPA3 SAE workflow

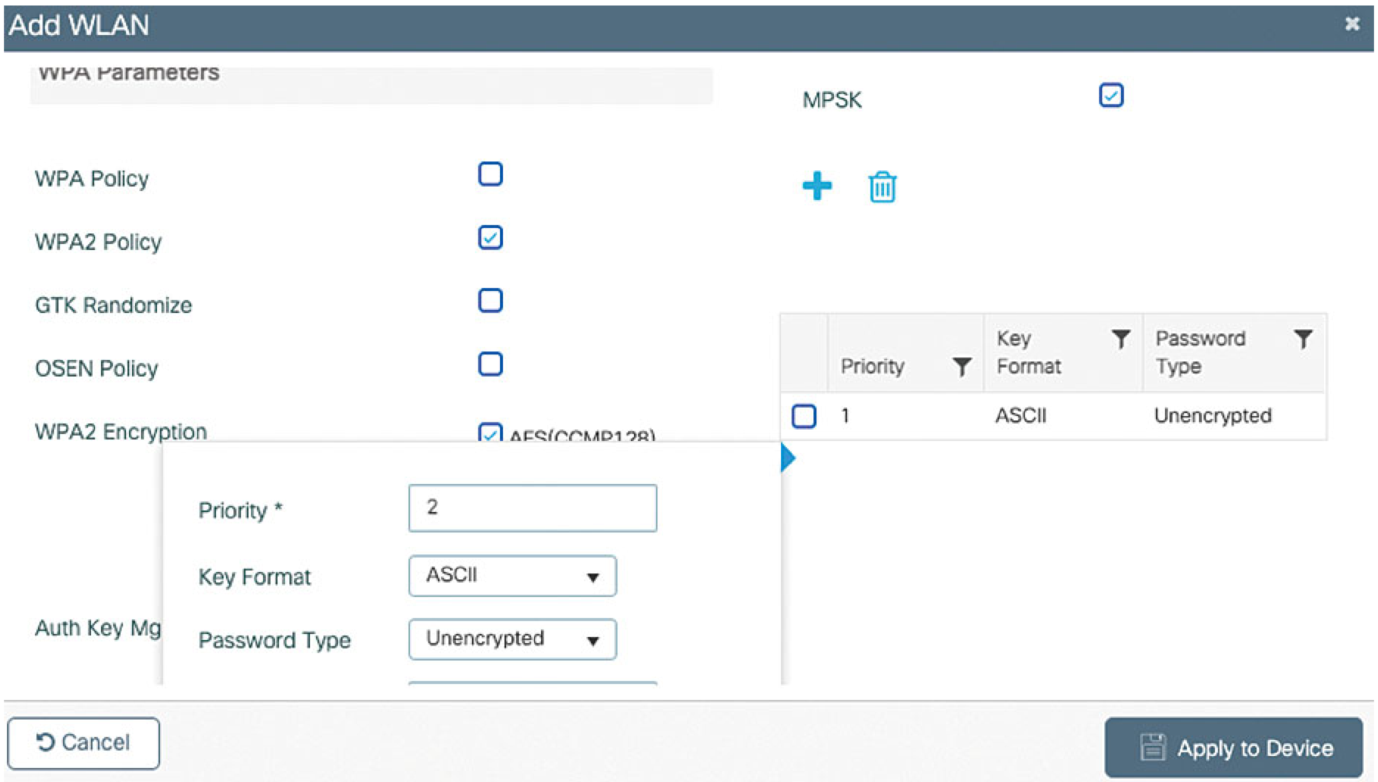


Figure 5-25 MPSK configuration in WLAN settings

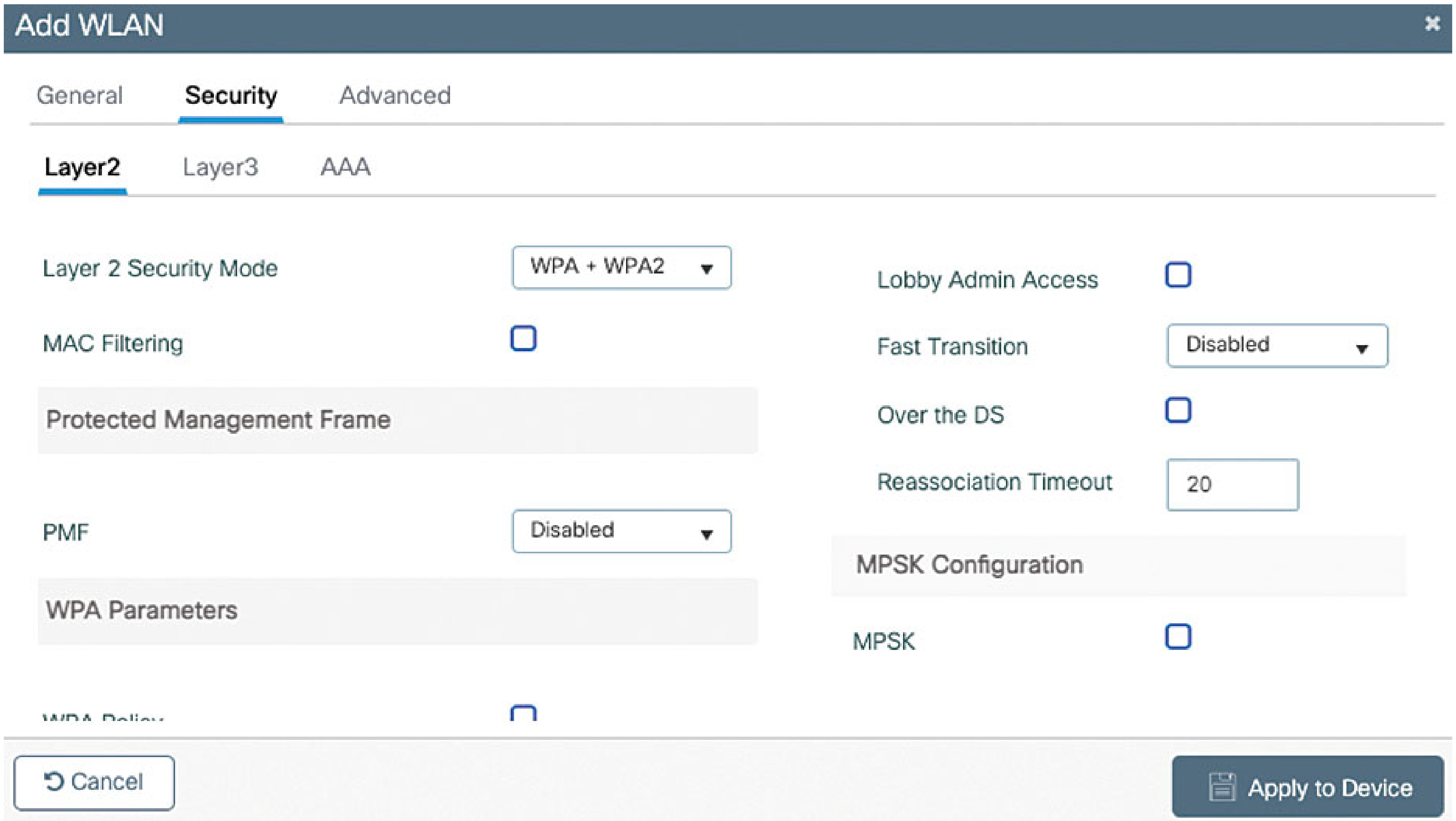


Figure 5-26 WPA 2 PSK SSID settings

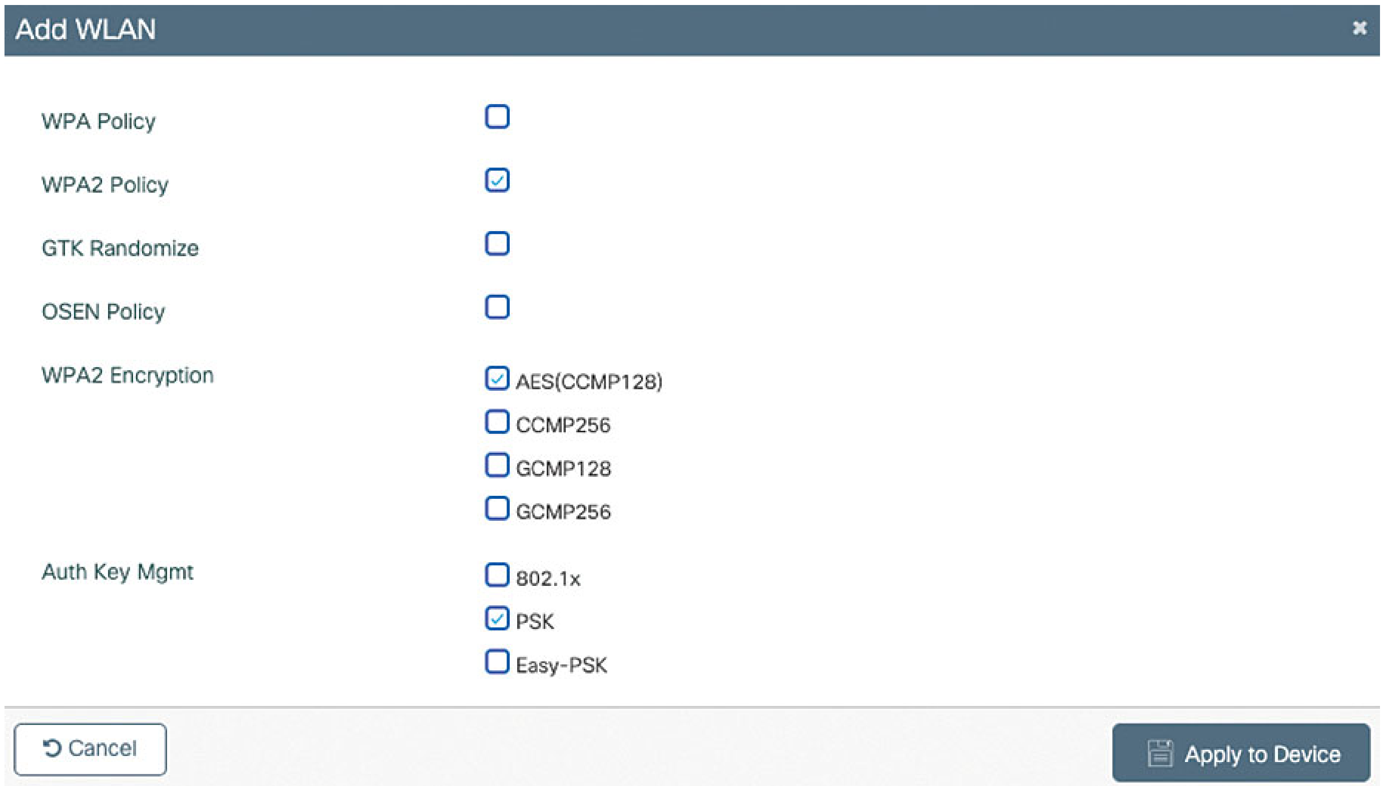


Figure 5-27 WPA2 PSK Layer2 security settings

Figure 5-28 WPA2 PSK SSID key configuration

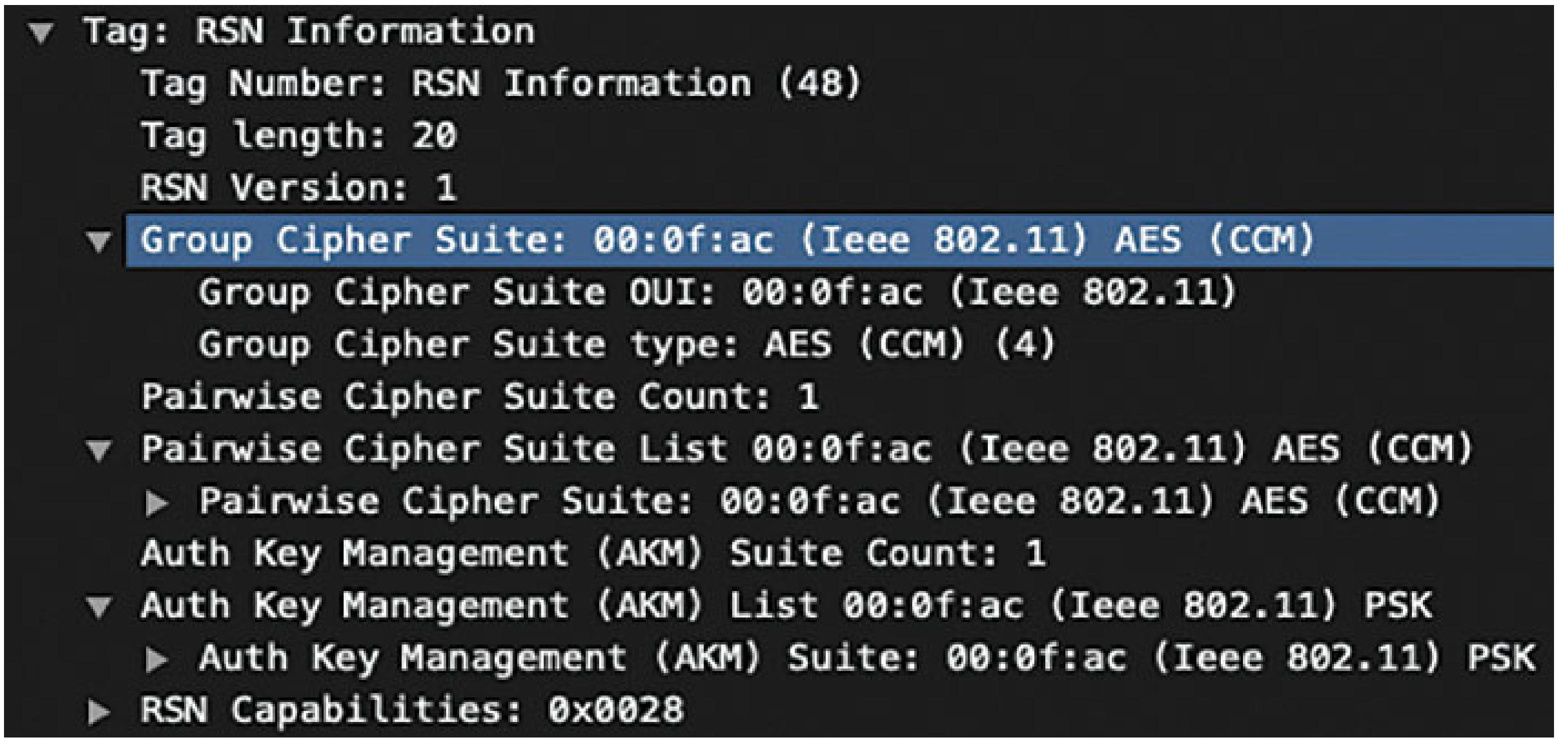


Figure 5-29 RSN information element of a beacon frame of a WPA PSK SSID

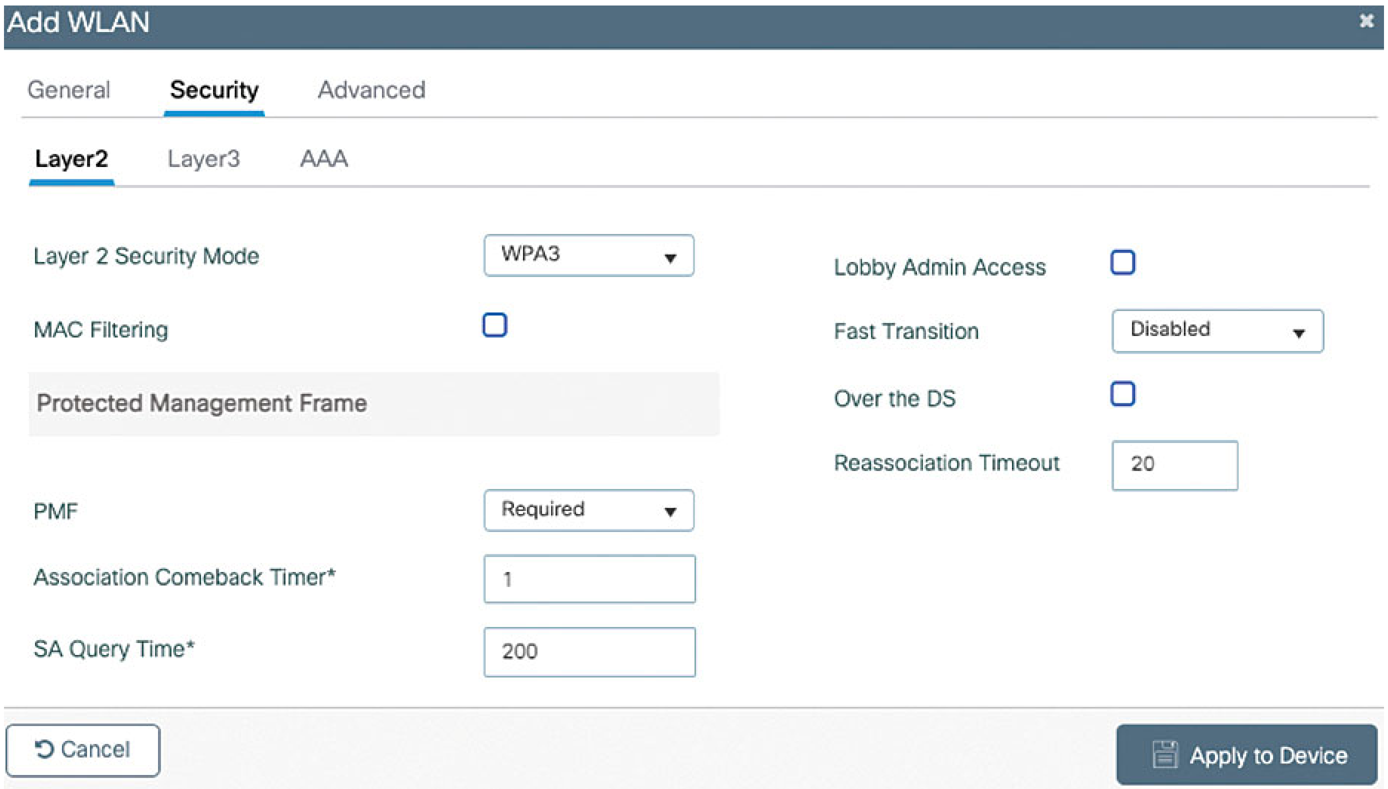


Figure 5-30 SAE WLAN example configuration

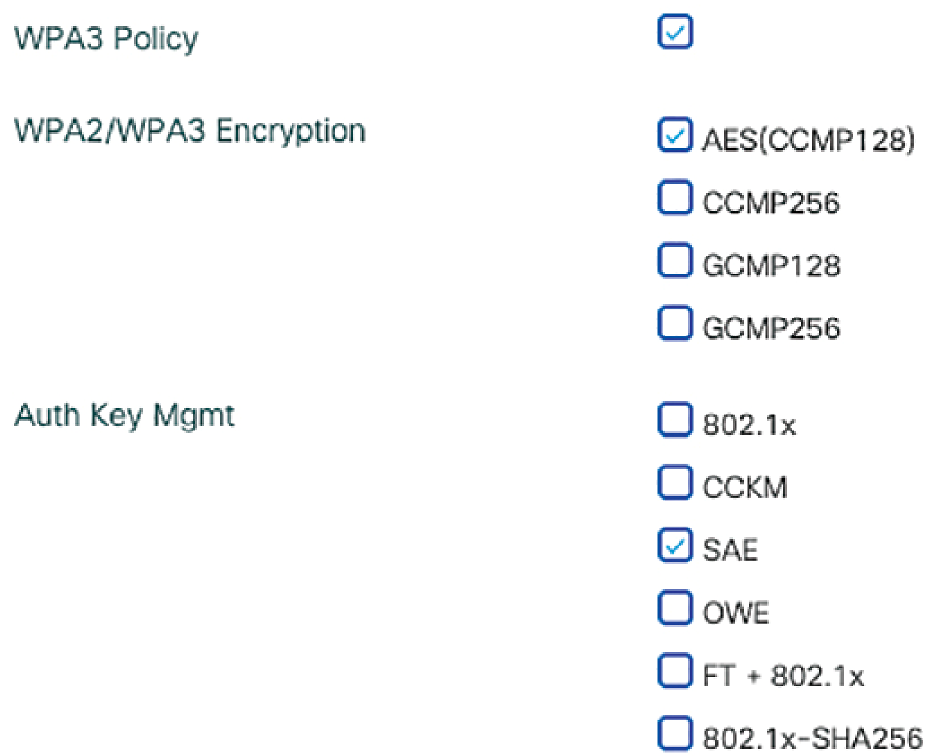


Figure 5-31 Ciphers option example for SAE

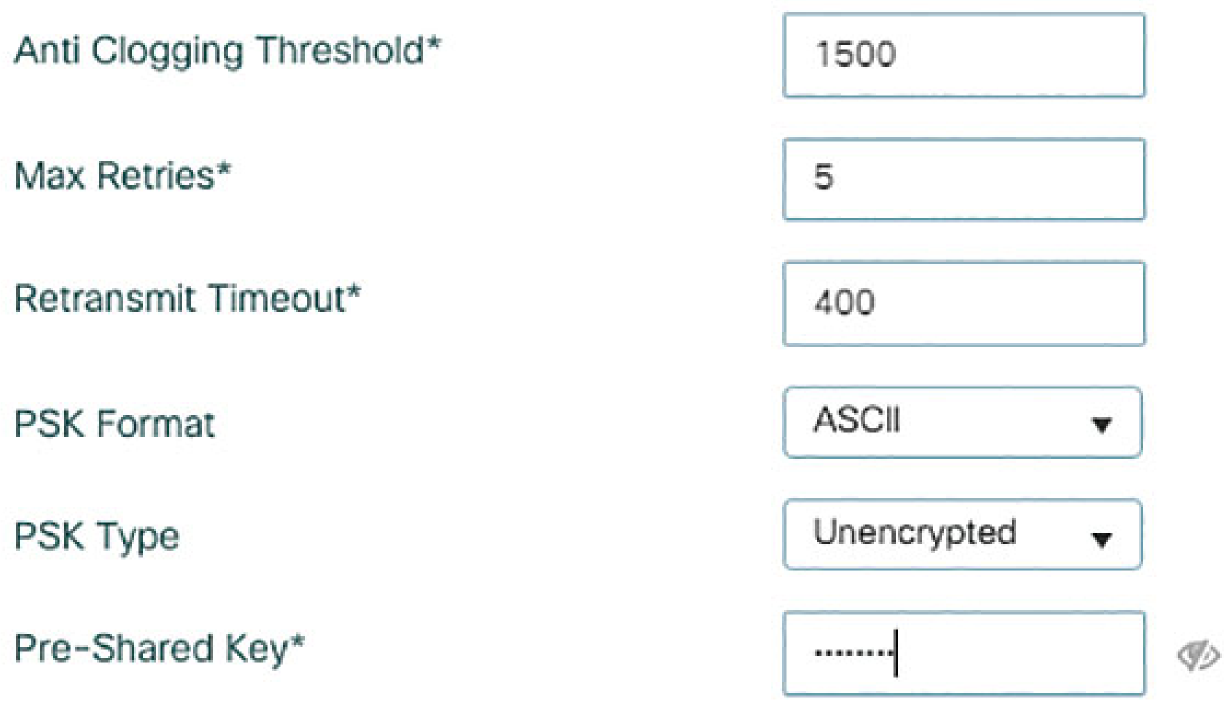


Figure 5-32 PSK configuration for SAE

```
▼ Tag: RSN Information
    Tag Number: RSN Information (48)
    Tag length: 26
    RSN Version: 1
  ▼ Group Cipher Suite: 00:0f:ac (Ieee 802.11) AES (CCM)
      Group Cipher Suite OUI: 00:0f:ac (Ieee 802.11)
      Group Cipher Suite type: AES (CCM) (4)
    Pairwise Cipher Suite Count: 1
  ▼ Pairwise Cipher Suite List 00:0f:ac (Ieee 802.11) AES (CCM)
    ▶ Pairwise Cipher Suite: 00:0f:ac (Ieee 802.11) AES (CCM)
    Auth Key Management (AKM) Suite Count: 1
  ▼ Auth Key Management (AKM) List 00:0f:ac (Ieee 802.11) SAE (SHA256)
    ▶ Auth Key Management (AKM) Suite: 00:0f:ac (Ieee 802.11) SAE (SHA256)
  ▶ RSN Capabilities: 0x00e8
    PMKID Count: 0
    PMKID List
  ▼ Group Management Cipher Suite: 00:0f:ac (Ieee 802.11) BIP (128)
      Group Management Cipher Suite OUI: 00:0f:ac (Ieee 802.11)
      Group Management Cipher Suite type: BIP (128) (6)
```

Figure 5-33 SAE advertised in the SSID beacon RSN IE

From *Understanding and Troubleshooting Cisco Catalyst 9800 Series Wireless Controller by* Simone Arena, Francisco Sedano Crippa, Nicolas Darchis and Sudha Katgeri(9780137492329) Cisco Press

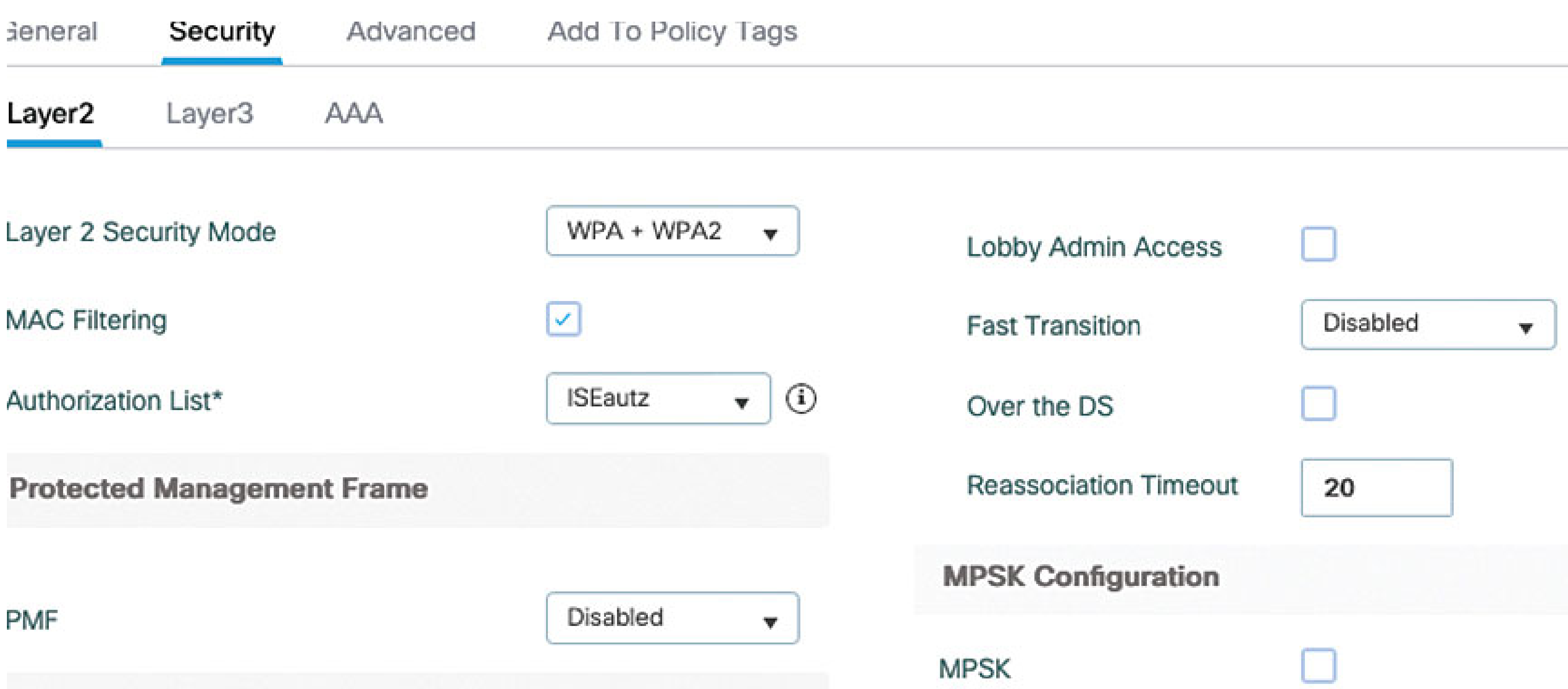


Figure 5-34 iPSK SSID security settings part 1

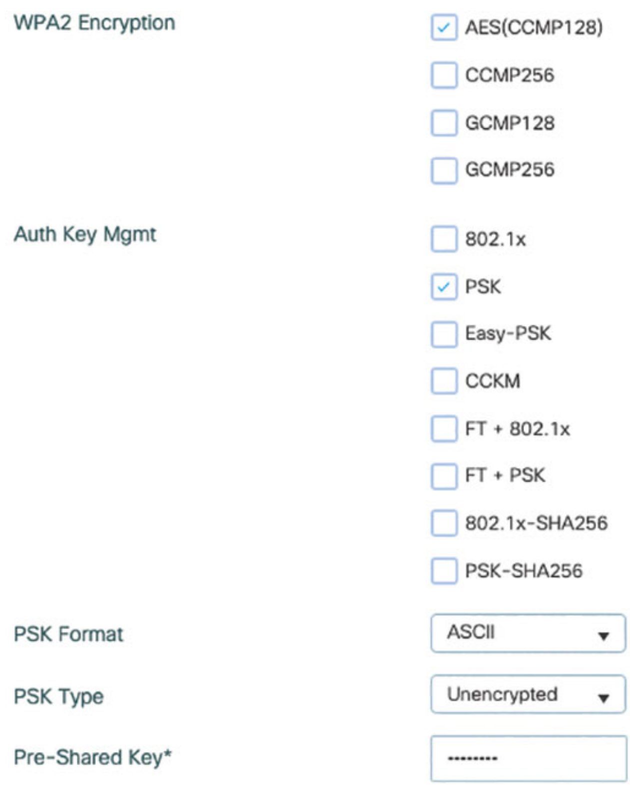


Figure 5-35 iPSK SSID security settings part 2

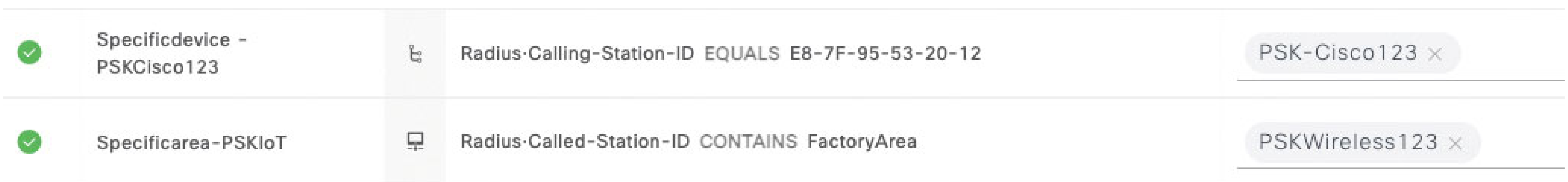


Figure 5-36 ISE authorization policies for IPSK

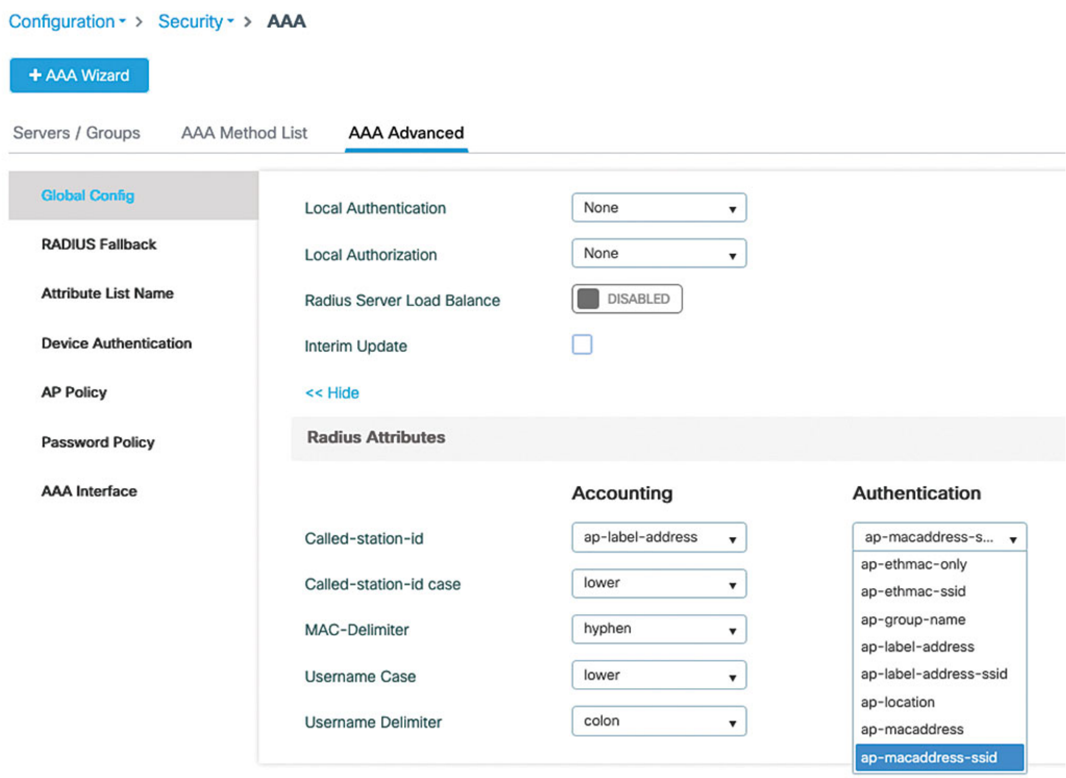


Figure 5-37 Customization of the called-station-id RADIUS field

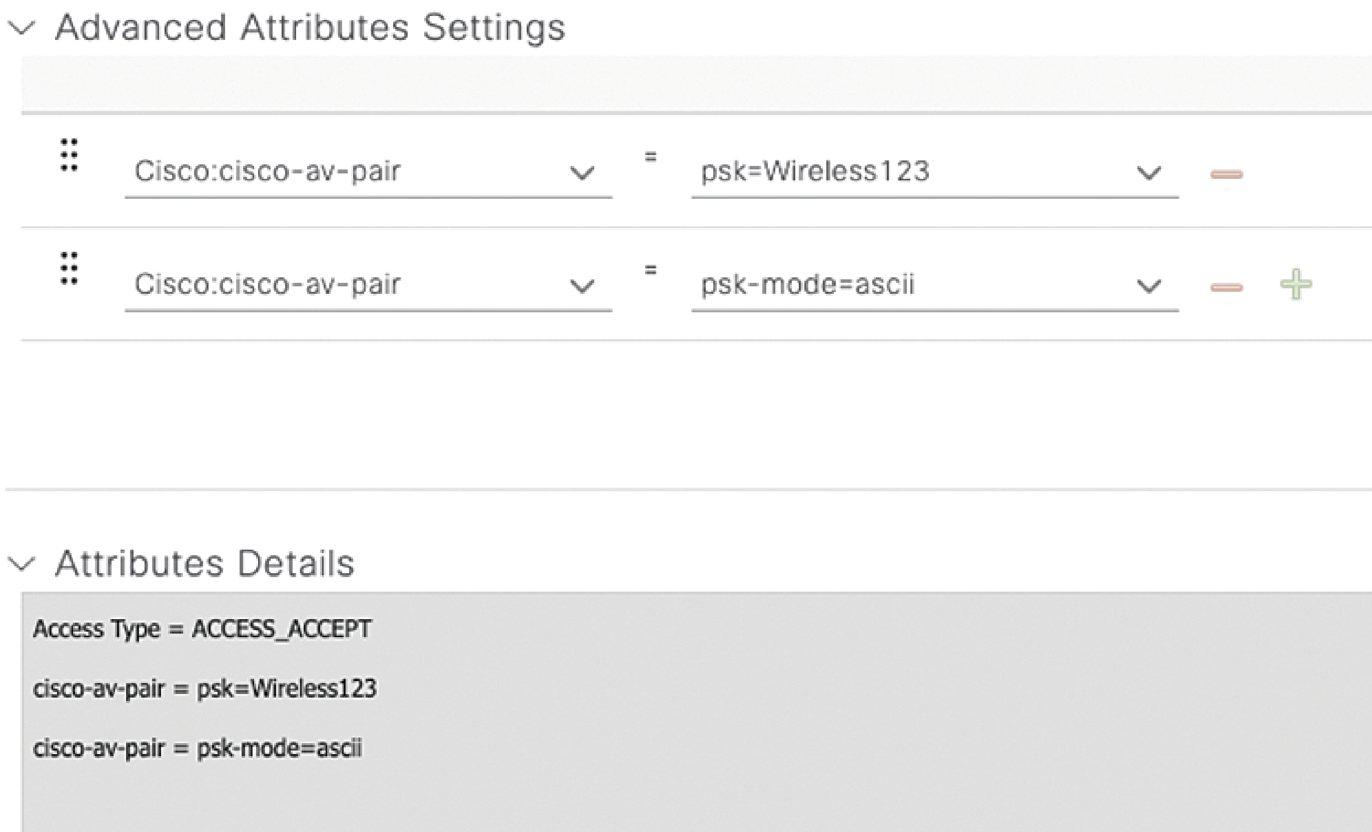


Figure 5-38 ISE authorization result

☐		EnhancedOpen		3	EnhancedOpen	[WPA3][OWE][AES]
☐		Open		4	Open	[open]

Figure 5-39 A pair of Open/Enhanced Open SSIDs for Enhanced Open Transition mode

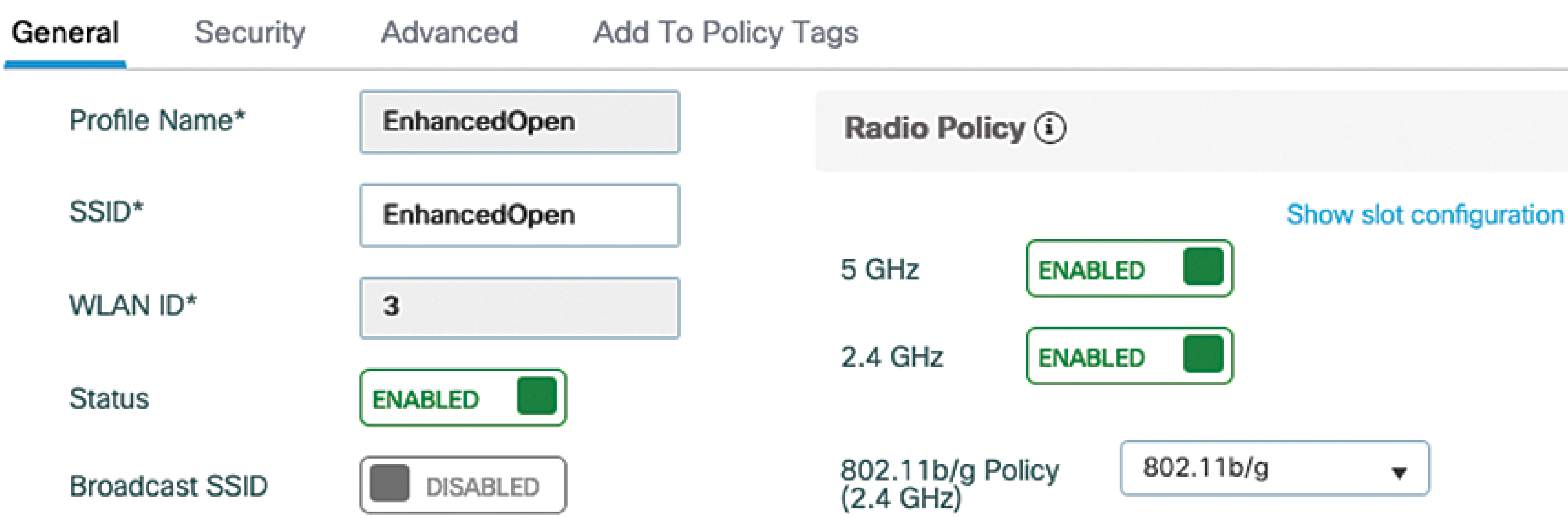


Figure 5-40 Enhanced Open SSID general settings

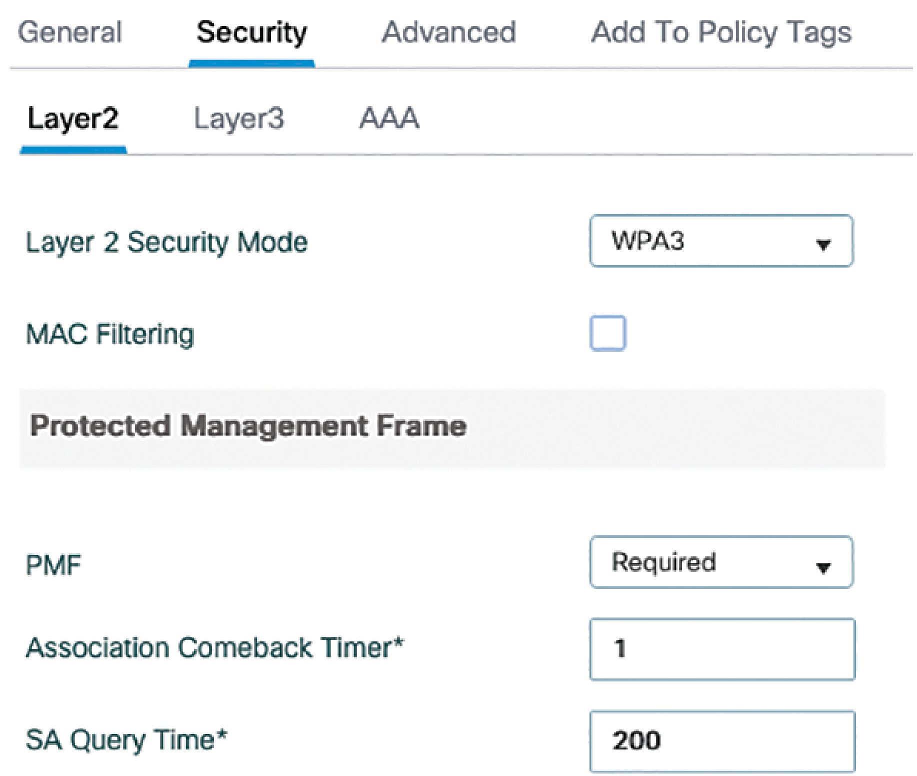


Figure 5-41 Enhanced Open security settings part 1

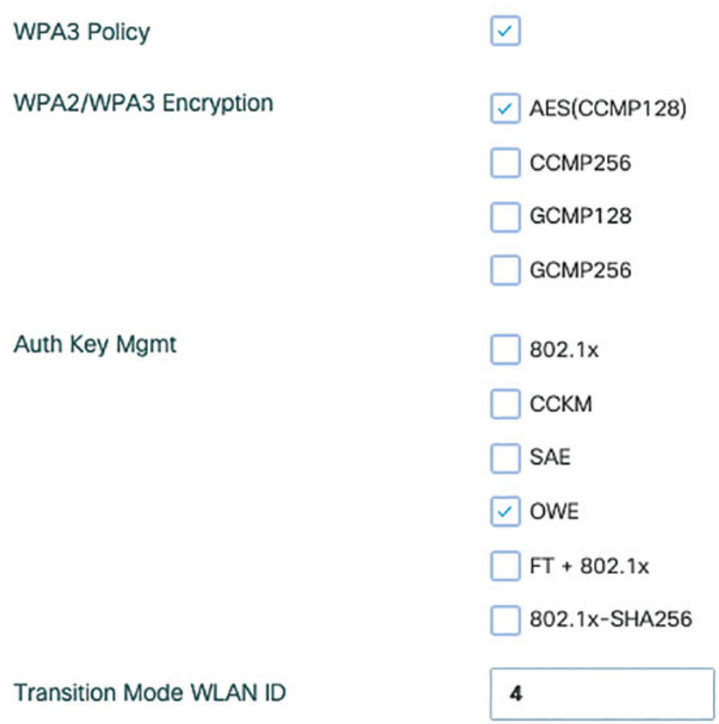


Figure 5-42 Enhanced Open SSID security settings part 2

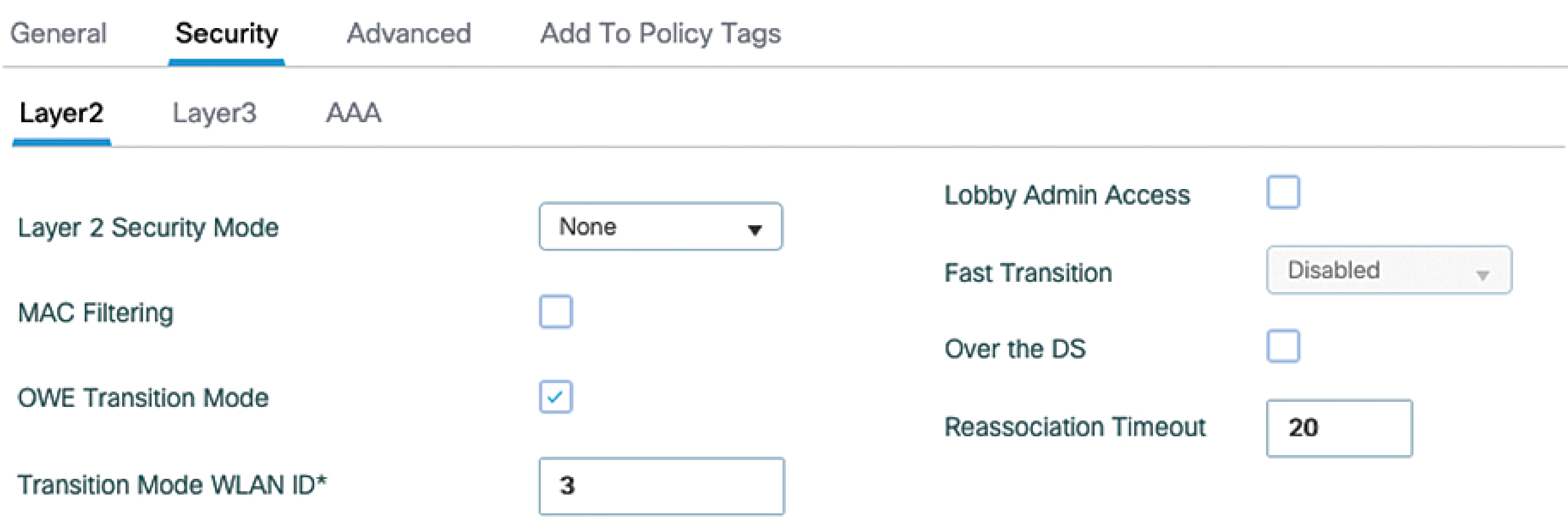


Figure 5-43 Regular open SSID linked with transition mode to an Enhanced Open SSID

WLAN Profile Name	EnhancedOpen
Wireless LAN Network Name (SSID)	EnhancedOpen
Client Entry Create Time	21 seconds
Policy Type	WPA3
Encryption Cipher	CCMP (AES)
Authentication Key Management	OWE

Figure 5-44 Security details of a client connected to an Enhanced Open SSID

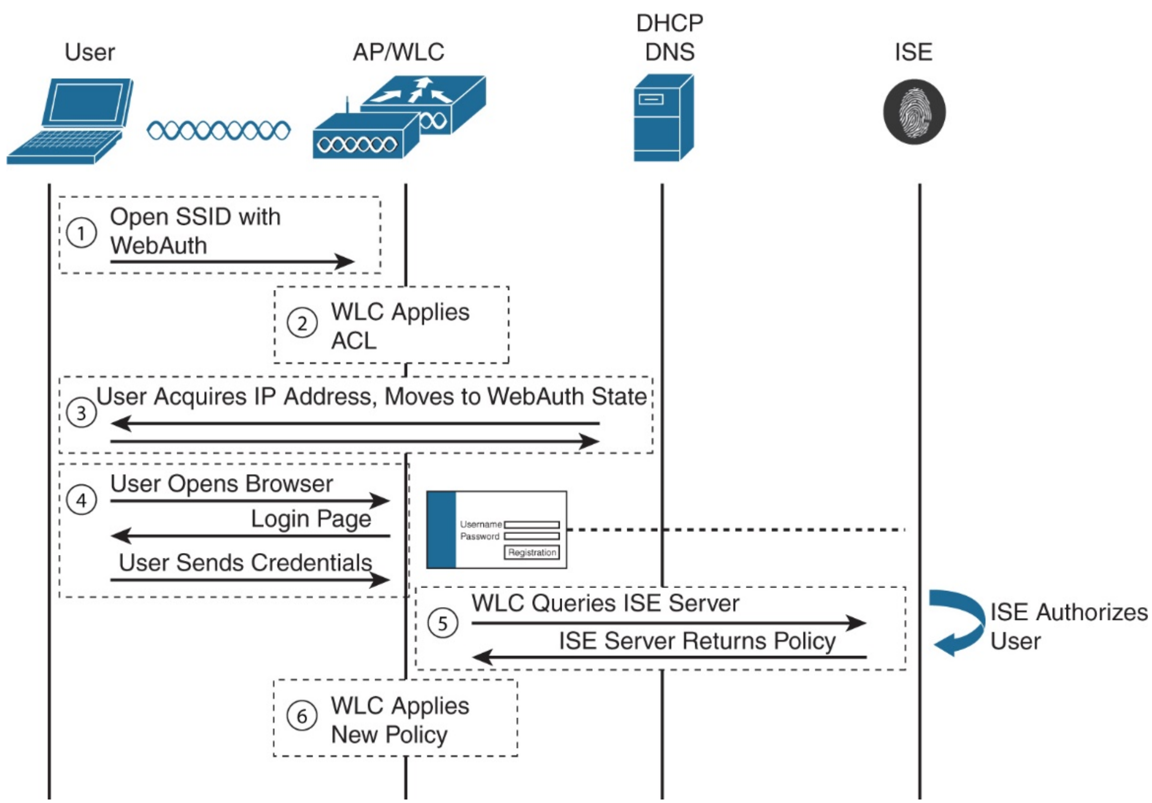


Figure 5-45 Local web authentication workflow

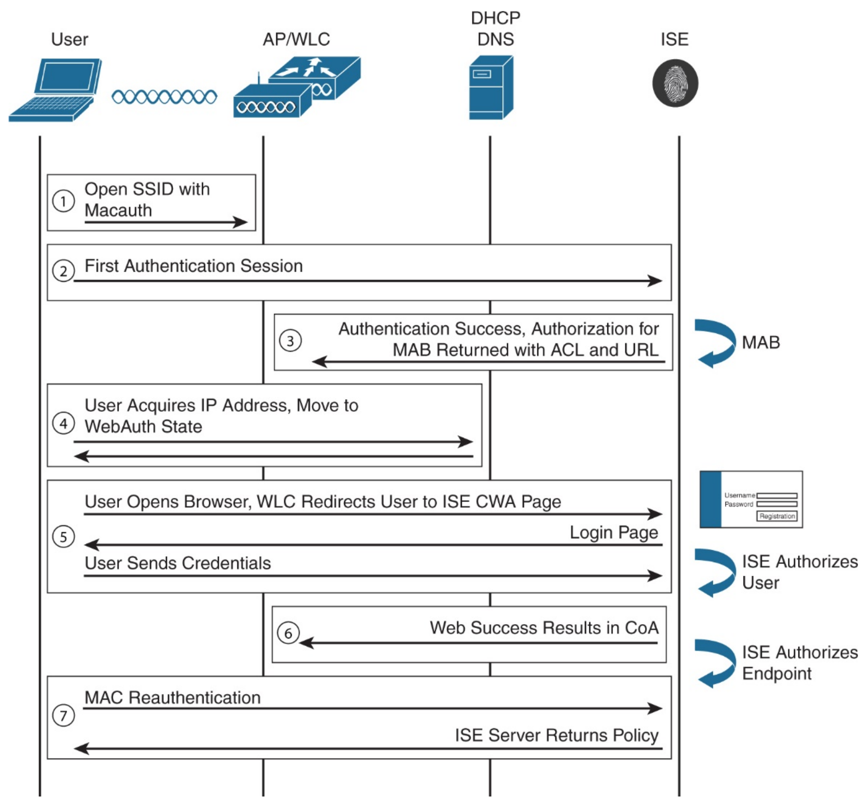


Figure 5-46 Central Web Authentication workflow

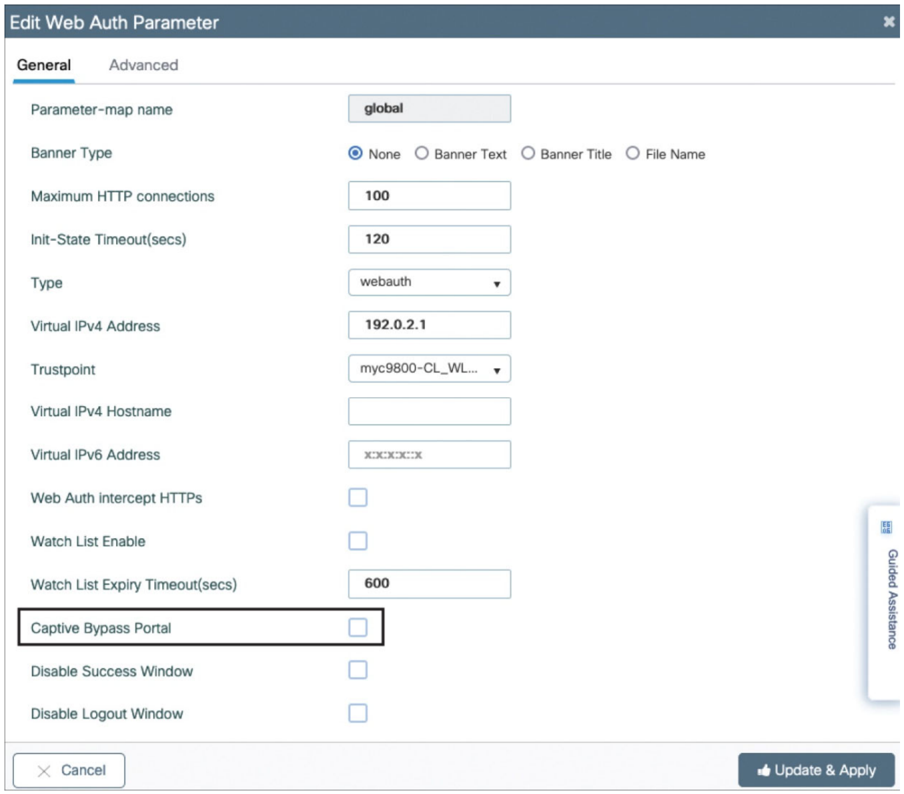


Figure 5-47 Do not enable Captive Bypass Portal unless you really have a good reason

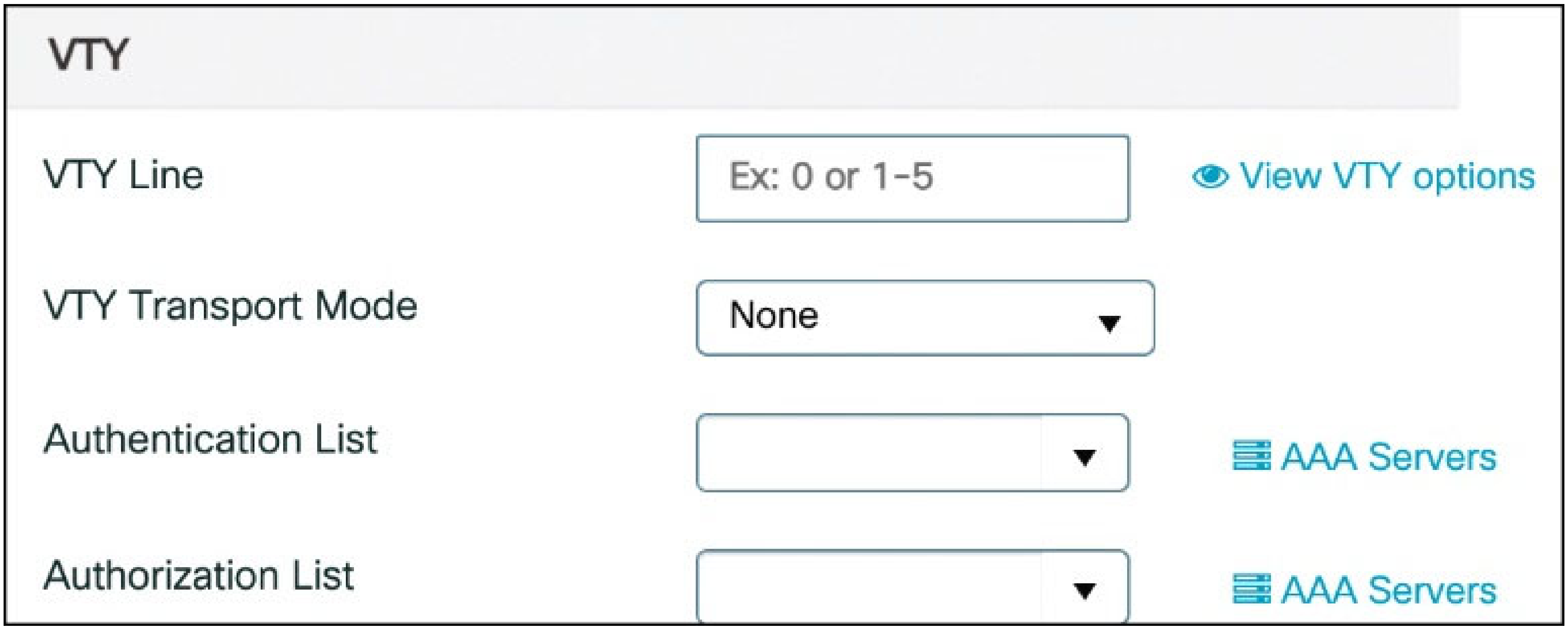


Figure 5-48 VTY configuration for CLI access in the HTTP/Netconf web page

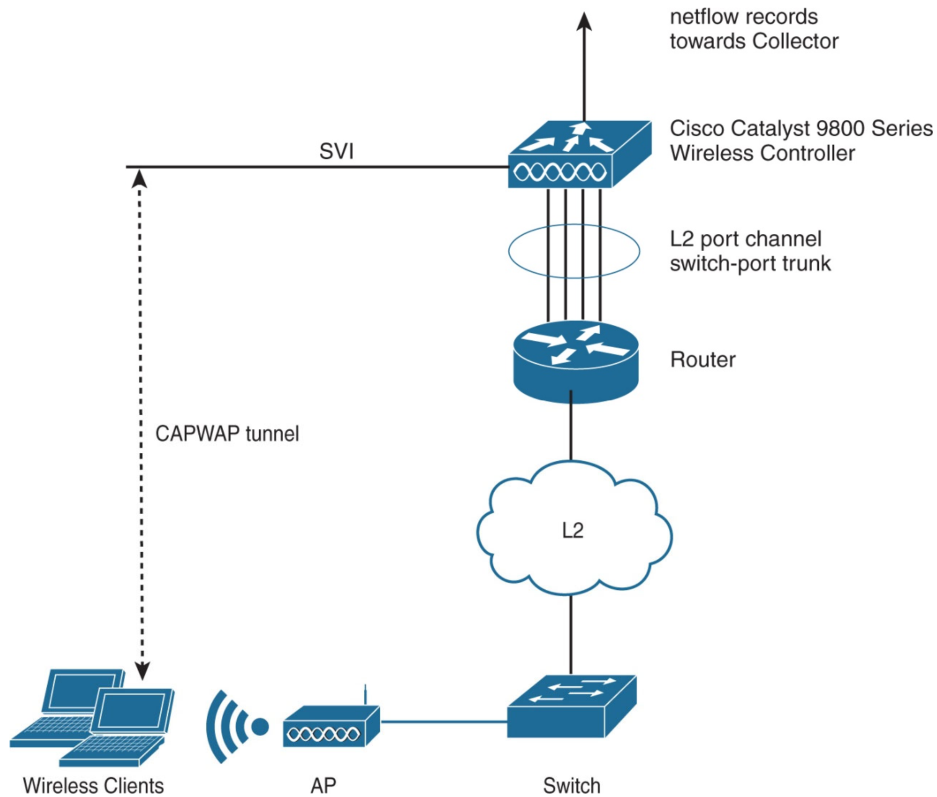


Figure 5-49 ETA workflow topology

Figure 5-50 Umbrella global configuration on the 9800

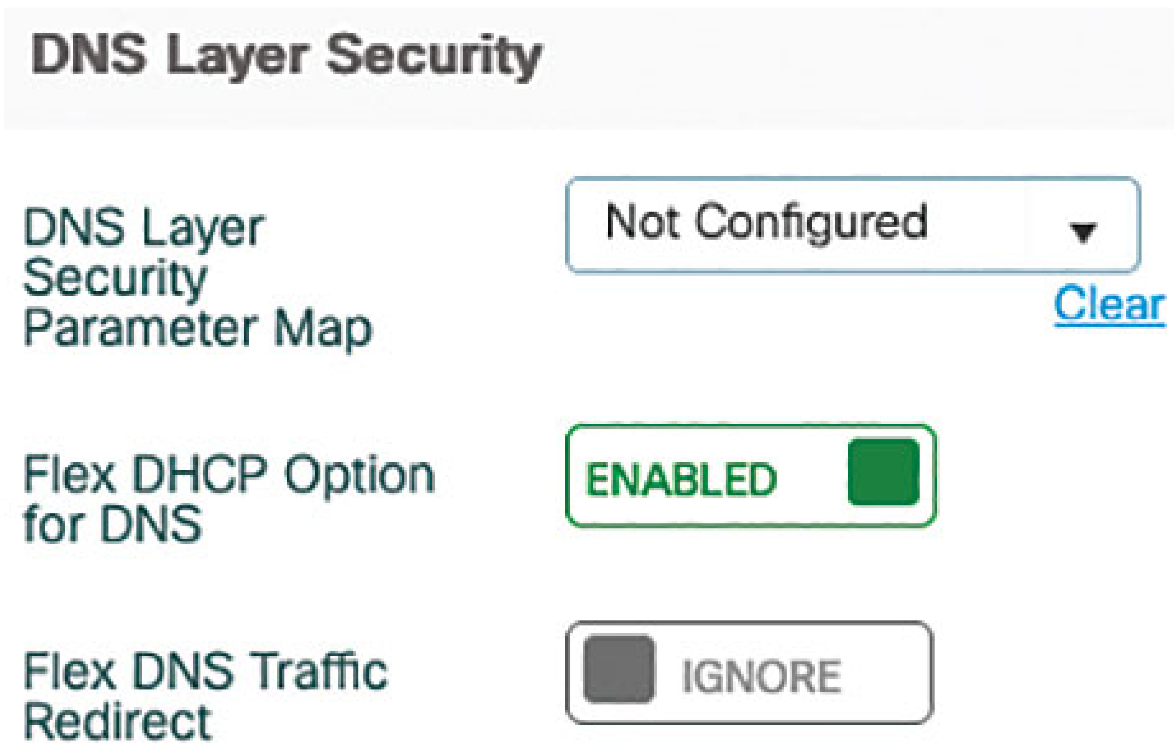


Figure 5-51 DNS Layer Security (formerly called Umbrella) section of a policy profile.

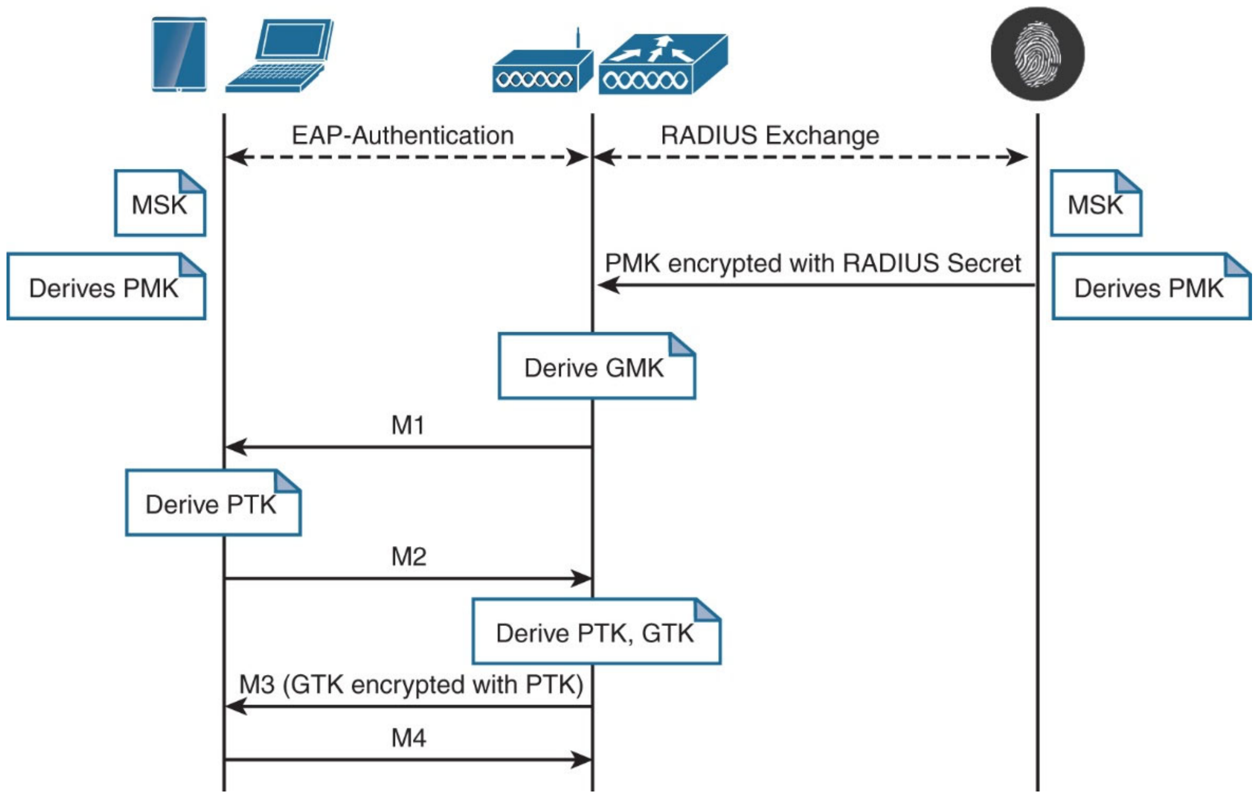


Figure 6-1 802.11 key management and distribution

No.	Time	Source	Destination	Length	Info
1716	0.022097	e2:2e:04:18:04:24	Cisco_9f:c3:2a	108	Authentication, SN=3813, FN=0, Flags=........C
1718	0.001447	Cisco_9f:c3:2a	e2:2e:04:18:04:24	108	Authentication, SN=0, FN=0, Flags=........C
1720	0.001086	e2:2e:04:18:04:24	Cisco_9f:c3:2a	361	Reassociation Request, SN=3814, FN=0, Flags=........C, SSID=slowroampsk
1723	0.002113	Cisco_9f:c3:2a	e2:2e:04:18:04:24	287	Reassociation Response, SN=1, FN=0, Flags=........C
1725	0.001293	Cisco_9f:c3:2a	e2:2e:04:18:04:24	221	Key (Message 1 of 4)
1731	0.020722	e2:2e:04:18:04:24	Cisco_9f:c3:2a	221	Key (Message 2 of 4)
1733	0.000607	Cisco_9f:c3:2a	e2:2e:04:18:04:24	255	Key (Message 3 of 4)
1735	0.012708	e2:2e:04:18:04:24	Cisco_9f:c3:2a	199	Key (Message 4 of 4)
1743	0.000052	e2:2e:04:18:04:24	Cisco_b1:6c:fd	144	QoS Data, SN=0, FN=0, Flags=.p.....TC

Figure 6-2 WPA/WPA2 PSK reassociation (slow roam)

No.	Time	Source	Destination	Protocol	Length	Info
4713	16:54:43.246686	Apple_64:5d:e5	68:7d:b4:5e:43:8f	802.11	69	Authentication, SN=2755, FN=0, Flags=........C
4715	16:54:43.248163	68:7d:b4:5e:43:8f	Apple_64:5d:e5	802.11	69	Authentication, SN=0, FN=0, Flags=........C
4717	16:54:43.248853	Apple_64:5d:e5	68:7d:b4:5e:43:8f	802.11	211	Reassociation Request, SN=2756, FN=0, Flags=........C, SSID=cvoice
4719	16:54:43.263550	68:7d:b4:5e:43:8f	Apple_64:5d:e5	802.11	183	Reassociation Response, SN=1, FN=0, Flags=........C
4721	16:54:43.266930	68:7d:b4:5e:43:8f	Apple_64:5d:e5	EAP	72	Request, Identity
4723	16:54:43.269581	Apple_64:5d:e5	68:7d:b4:5e:43:8f	EAP	76	Response, Identity
4725	16:54:43.276763	68:7d:b4:5e:43:8f	Apple_64:5d:e5	EAP	73	Request, TLS EAP (EAP-TLS)
4727	16:54:43.278065	Apple_64:5d:e5	68:7d:b4:5e:43:8f	EAP	75	Response, Legacy Nak (Response Only)
4729	16:54:43.284187	68:7d:b4:5e:43:8f	Apple_64:5d:e5	EAP	73	Request, Protected EAP (EAP-PEAP)
4731	16:54:43.285117	Apple_64:5d:e5	68:7d:b4:5e:43:8f	TLSv1.2	260	Client Hello
4736	16:54:43.302692	68:7d:b4:5e:43:8f	Apple_64:5d:e5	EAP	1079	Request, Protected EAP (EAP-PEAP)
4738	16:54:43.303598	Apple_64:5d:e5	68:7d:b4:5e:43:8f	EAP	73	Response, Protected EAP (EAP-PEAP)
4740	16:54:43.307753	68:7d:b4:5e:43:8f	Apple_64:5d:e5	TLSv1.2	351	Server Hello, Certificate, Server Key Exchange, Server Hello Done
4742	16:54:43.317178	Apple_64:5d:e5	68:7d:b4:5e:43:8f	TLSv1.2	203	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
4744	16:54:43.323470	68:7d:b4:5e:43:8f	Apple_64:5d:e5	TLSv1.2	124	Change Cipher Spec, Encrypted Handshake Message
4746	16:54:43.324119	Apple_64:5d:e5	68:7d:b4:5e:43:8f	EAP	73	Response, Protected EAP (EAP-PEAP)
4748	16:54:43.328433	68:7d:b4:5e:43:8f	Apple_64:5d:e5	TLSv1.2	107	Application Data
4750	16:54:43.329524	Apple_64:5d:e5	68:7d:b4:5e:43:8f	TLSv1.2	111	Application Data
4752	16:54:43.333737	68:7d:b4:5e:43:8f	Apple_64:5d:e5	TLSv1.2	137	Application Data
4754	16:54:43.334653	Apple_64:5d:e5	68:7d:b4:5e:43:8f	TLSv1.2	165	Application Data
4756	16:54:43.346834	68:7d:b4:5e:43:8f	Apple_64:5d:e5	TLSv1.2	153	Application Data
4758	16:54:43.347552	Apple_64:5d:e5	68:7d:b4:5e:43:8f	TLSv1.2	108	Application Data
4760	16:54:43.351714	68:7d:b4:5e:43:8f	Apple_64:5d:e5	TLSv1.2	106	Application Data
4762	16:54:43.352310	Apple_64:5d:e5	68:7d:b4:5e:43:8f	EAP	73	Response, Protected EAP (EAP-PEAP)
4764	16:54:43.366697	68:7d:b4:5e:43:8f	Apple_64:5d:e5	EAP	71	Success
4766	16:54:43.367407	68:7d:b4:5e:43:8f	Apple_64:5d:e5	EAPOL	184	Key (Message 1 of 4)
4768	16:54:43.368210	Apple_64:5d:e5	68:7d:b4:5e:43:8f	EAPOL	202	Key (Message 2 of 4)
4770	16:54:43.370090	68:7d:b4:5e:43:8f	Apple_64:5d:e5	EAPOL	274	Key (Message 3 of 4)
4772	16:54:43.370773	Apple_64:5d:e5	68:7d:b4:5e:43:8f	EAPOL	162	Key (Message 4 of 4)

Figure 6-3 WPA/WPA2 EAP reassociation (slow roam)

No.	Time	Source	Destination	Length	Info
554	0.002172	92:4c:d7:78:89:f8	Cisco_23:c6:4f	108	Authentication, SN=1823, FN=0, Flags=........C
556	0.000992	Cisco_23:c6:4f	92:4c:d7:78:89:f8	108	Authentication, SN=3128, FN=0, Flags=........C
558	0.000985	92:4c:d7:78:89:f8	Cisco_23:c6:4f	347	Association Request, SN=1824, FN=0, Flags=........C, SSID=OKC
559	0.000136	92:4c:d7:78:89:f8	Cisco_23:c6:4f	347	Association Request, SN=1824, FN=0, Flags=....R...C, SSID=OKC
561	0.004662	Cisco_23:c6:4f	92:4c:d7:78:89:f8	269	Association Response, SN=3129, FN=0, Flags=........C
563	0.027169	Cisco_23:c6:4f	92:4c:d7:78:89:f8	109	Request, Identity
569	0.008778	92:4c:d7:78:89:f8	Cisco_23:c6:4f	115	Response, Identity
573	0.021785	Cisco_23:c6:4f	92:4c:d7:78:89:f8	110	Request, Protected EAP (EAP-PEAP)
575	0.006260	92:4c:d7:78:89:f8	Cisco_23:c6:4f	241	Client Hello
599	0.000000	Cisco_23:c6:4f	92:4c:d7:78:89:f8	101	Request, Protected EAP (EAP-PEAP)
602	0.001996	92:4c:d7:78:89:f8	Cisco_23:c6:4f	110	Response, Protected EAP (EAP-PEAP)
604	0.003478	Cisco_23:c6:4f	92:4c:d7:78:89:f8	1321	Server Hello, Certificate, Server Key Exchange, Server Hello Done
606	0.007491	92:4c:d7:78:89:f8	Cisco_23:c6:4f	236	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
608	0.001755	Cisco_23:c6:4f	92:4c:d7:78:89:f8	165	Change Cipher Spec, Encrypted Handshake Message
610	0.002876	92:4c:d7:78:89:f8	Cisco_23:c6:4f	110	Response, Protected EAP (EAP-PEAP)
612	0.001465	Cisco_23:c6:4f	92:4c:d7:78:89:f8	144	Application Data
614	0.002854	92:4c:d7:78:89:f8	Cisco_23:c6:4f	150	Application Data
616	0.001802	Cisco_23:c6:4f	92:4c:d7:78:89:f8	144	Application Data
618	0.002206	92:4c:d7:78:89:f8	Cisco_23:c6:4f	152	Application Data
620	0.001553	Cisco_23:c6:4f	92:4c:d7:78:89:f8	150	Application Data
622	0.002853	92:4c:d7:78:89:f8	Cisco_23:c6:4f	150	Application Data
624	0.001847	Cisco_23:c6:4f	92:4c:d7:78:89:f8	108	Success
626	0.000466	Cisco_23:c6:4f	92:4c:d7:78:89:f8	221	Key (Message 1 of 4)
628	0.007728	92:4c:d7:78:89:f8	Cisco_23:c6:4f	221	Key (Message 2 of 4)
630	0.001235	Cisco_23:c6:4f	92:4c:d7:78:89:f8	255	Key (Message 3 of 4)
633	0.000585	92:4c:d7:78:89:f8	Cisco_23:c6:4f	199	Key (Message 4 of 4)
634	0.000000	92:4c:d7:78:89:f8	Cisco_23:c6:4f	199	Key (Message 4 of 4)
2026	0.022106	92:4c:d7:78:89:f8	Cisco_9f:c3:2f	108	Authentication, SN=2190, FN=0, Flags=........C
2028	0.001362	Cisco_9f:c3:2f	92:4c:d7:78:89:f8	108	Authentication, SN=0, FN=0, Flags=........C
2030	0.001320	92:4c:d7:78:89:f8	Cisco_9f:c3:2f	371	Reassociation Request, SN=2191, FN=0, Flags=........C, SSID=OKC
2033	0.003157	Cisco_9f:c3:2f	92:4c:d7:78:89:f8	287	Reassociation Response, SN=1, FN=0, Flags=........C
2035	0.001324	Cisco_9f:c3:2f	92:4c:d7:78:89:f8	221	Key (Message 1 of 4)
2043	0.034286	92:4c:d7:78:89:f8	Cisco_9f:c3:2f	239	Key (Message 2 of 4)
2045	0.001272	Cisco_9f:c3:2f	92:4c:d7:78:89:f8	255	Key (Message 3 of 4)
2048	0.000248	92:4c:d7:78:89:f8	Cisco_9f:c3:2f	199	Key (Message 4 of 4)

Figure 6-4 Packet capture: OKC fast roam in local mode deployment

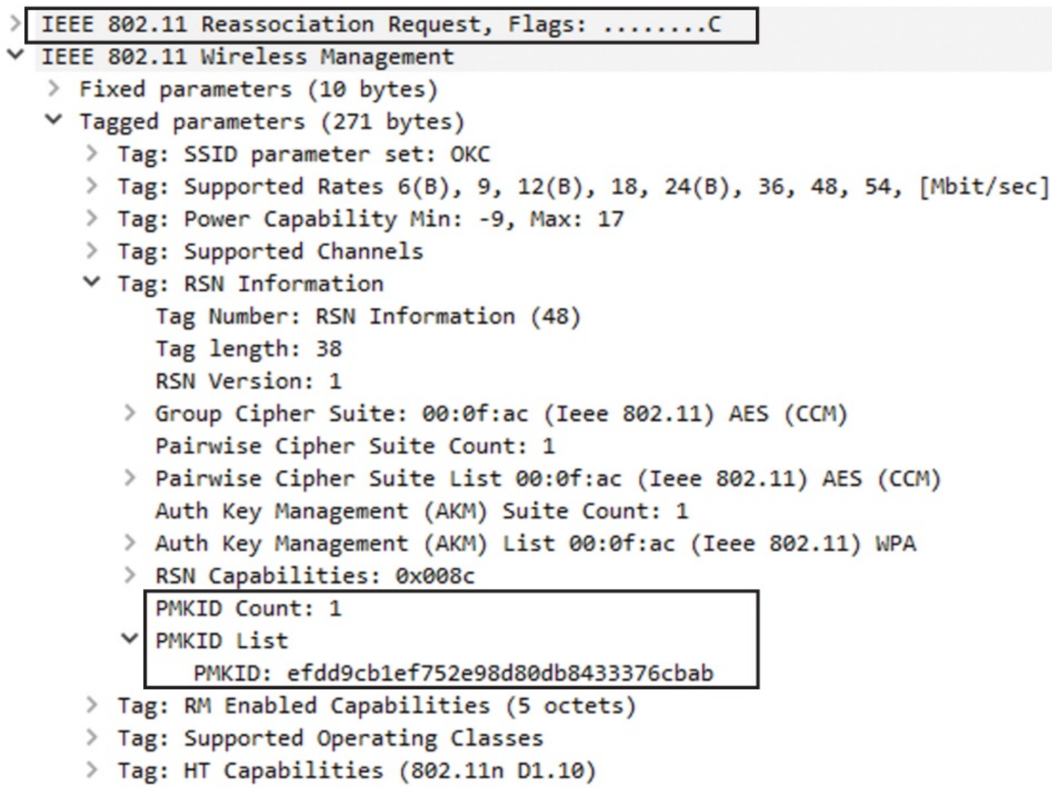


Figure 6-5 OKC fast roam RSN IE

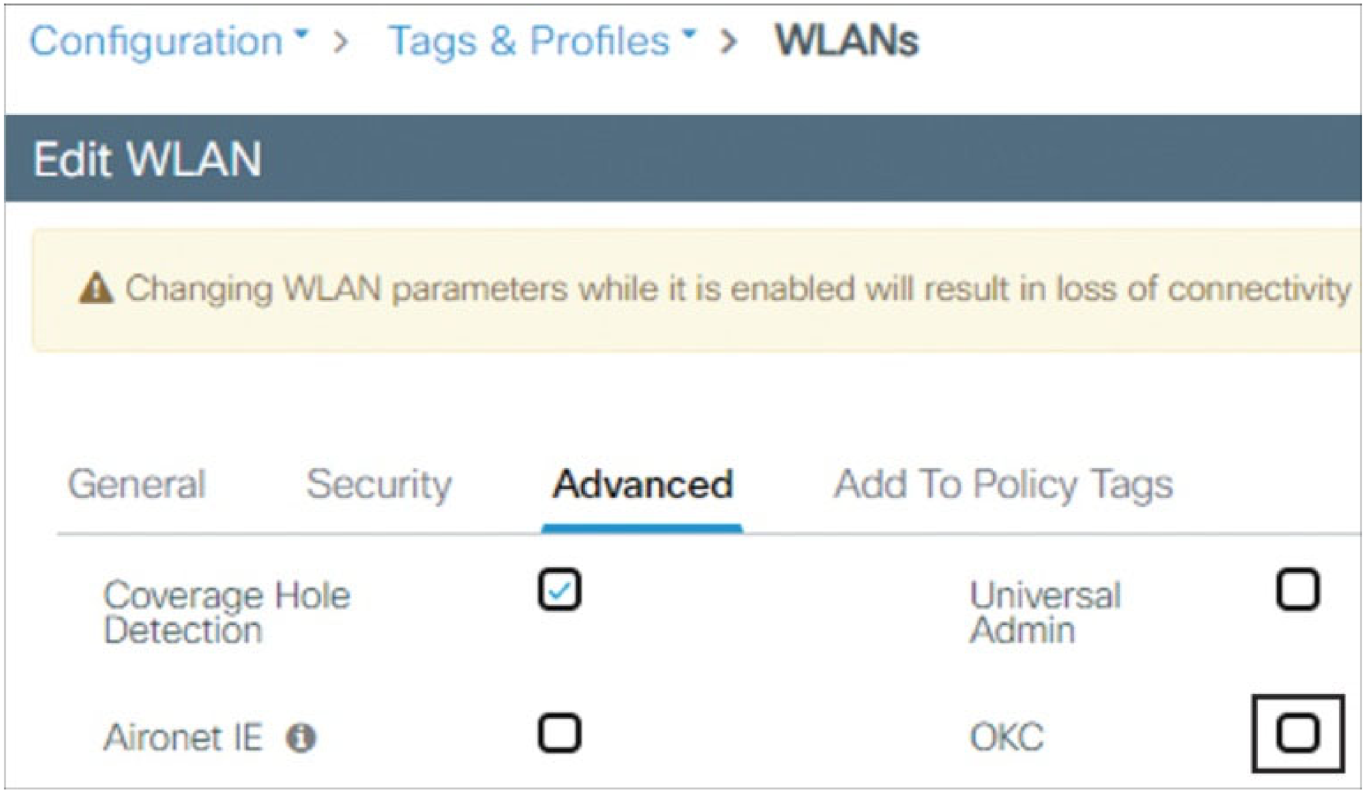


Figure 6-6 Disabling OKC for FlexConnect local authorization

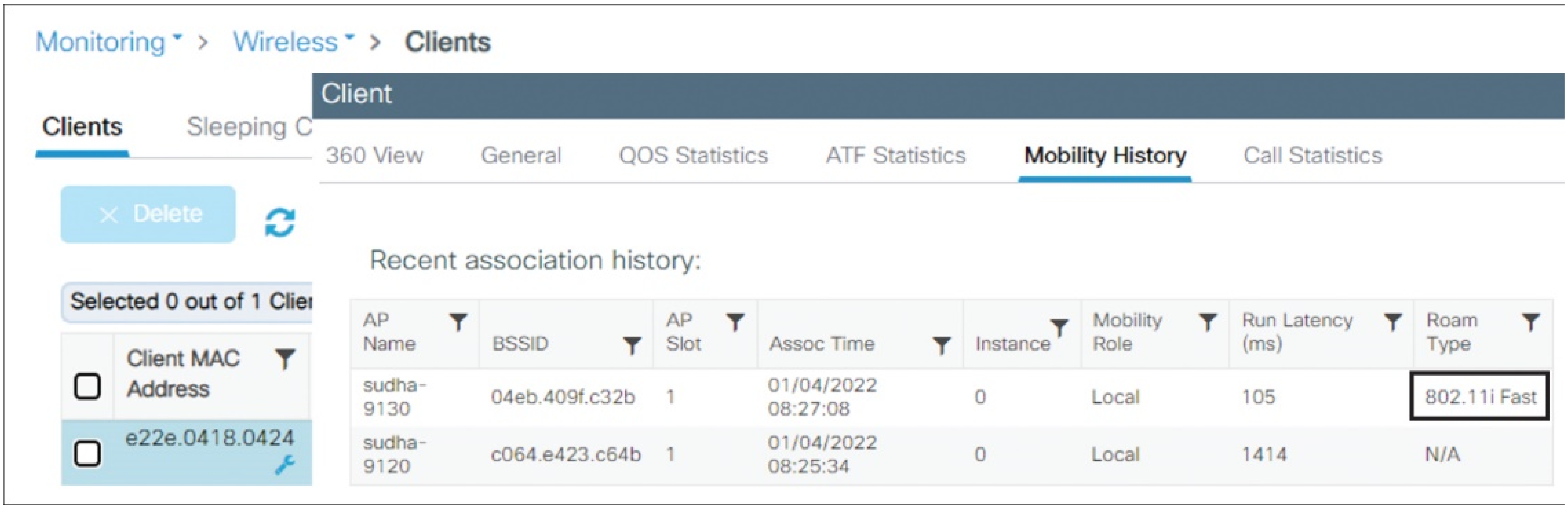


Figure 6-7 Monitoring the client roam type on the C9800

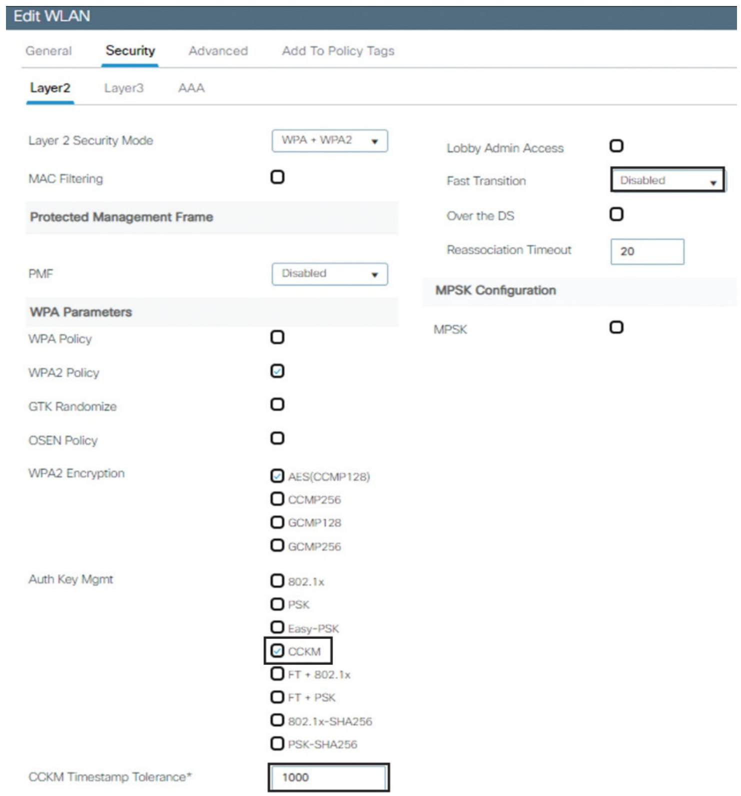


Figure 6-8 Configuring CCKM on the C9800

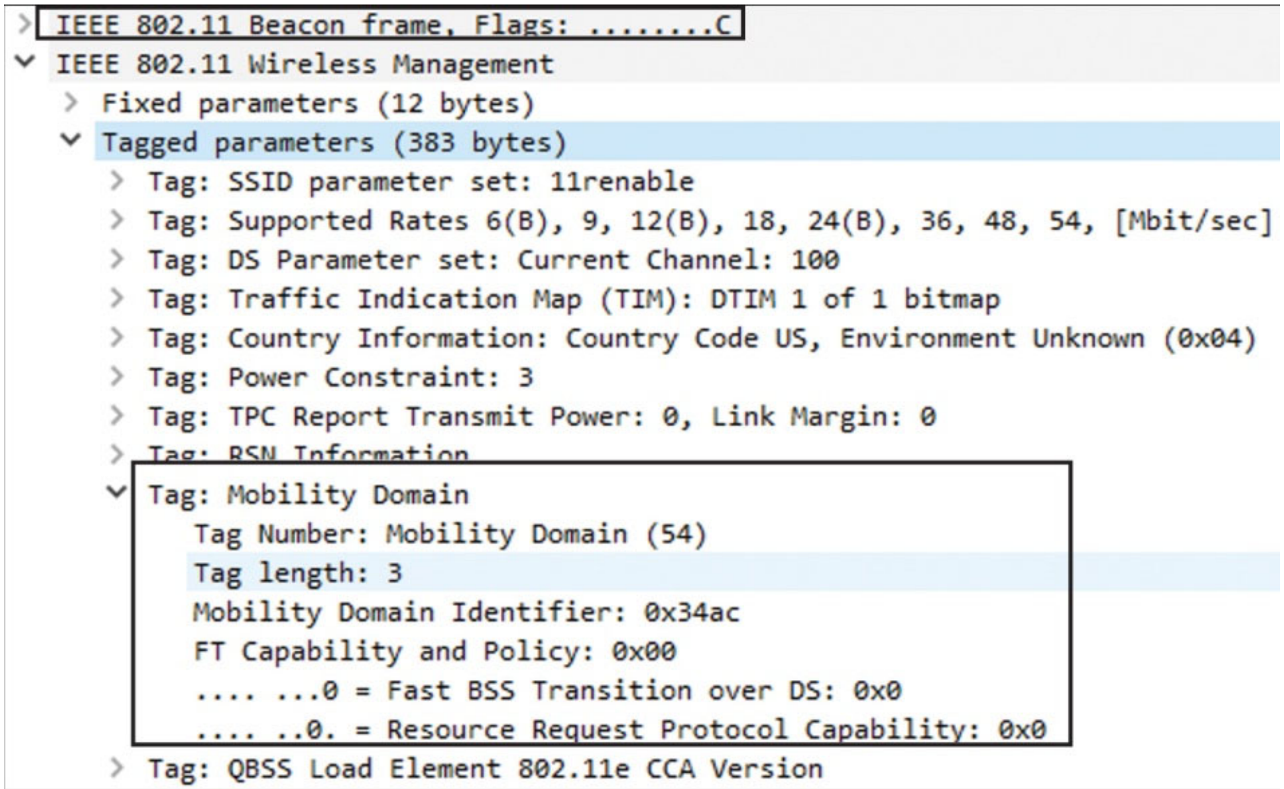


Figure 6-9 802.11r FT Beacon MDIE

```
> IEEE 802.11 Beacon frame, Flags: ........C
v IEEE 802.11 Wireless Management
  > Fixed parameters (12 bytes)
  v Tagged parameters (383 bytes)
    > Tag: SSID parameter set: 11renable
    > Tag: Supported Rates 6(B), 9, 12(B), 18, 24(B), 36, 48, 54, [Mbit/sec]
    > Tag: DS Parameter set: Current Channel: 100
    > Tag: Traffic Indication Map (TIM): DTIM 1 of 1 bitmap
    > Tag: Country Information: Country Code US, Environment Unknown (0x04)
    > Tag: Power Constraint: 3
    > Tag: TPC Report Transmit Power: 0, Link Margin: 0
    v Tag: RSN Information
        Tag Number: RSN Information (48)
        Tag length: 20
        RSN Version: 1
      > Group Cipher Suite: 00:0f:ac (Ieee 802.11) AES (CCM)
        Pairwise Cipher Suite Count: 1
      > Pairwise Cipher Suite List 00:0f:ac (Ieee 802.11) AES (CCM)
        Auth Key Management (AKM) Suite Count: 1
      v Auth Key Management (AKM) List 00:0f:ac (Ieee 802.11) FT over IEEE 802.1X
        v Auth Key Management (AKM) Suite: 00:0f:ac (Ieee 802.11) FT over IEEE 802.1X
            Auth Key Management (AKM) OUI: 00:0f:ac (Ieee 802.11)
            Auth Key Management (AKM) type: FT over IEEE 802.1X (3)
      > RSN Capabilities: 0x0028
    > Tag: Mobility Domain
```

Figure 6-10 802.11r FT Beacon RSN IE AKM

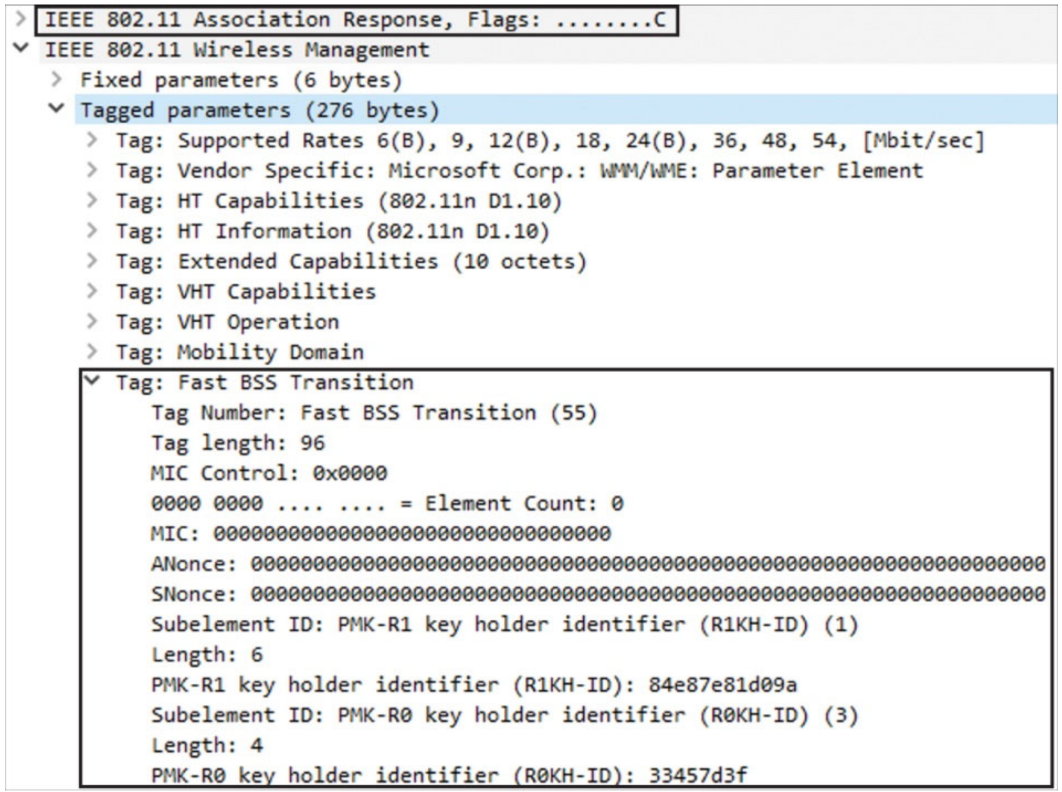


Figure 6-11 802.11r FT association response

No.	Time	Source	Destination	Length	Info
285	0.002251	MurataMa_60:29:3e	Cisco_23:c6:4c	108	Authentication, SN=271, FN=0, Flags=........C
287	0.001038	Cisco_23:c6:4c	MurataMa_60:29:3e	108	Authentication, SN=938, FN=0, Flags=........C
289	0.001673	MurataMa_60:29:3e	Cisco_23:c6:4c	358	Association Request, SN=272, FN=0, Flags=........C, SSID=11renable
291	0.004228	Cisco_23:c6:4c	MurataMa_60:29:3e	372	Association Response, SN=939, FN=0, Flags=........C
293	0.004293	Cisco_23:c6:4c	MurataMa_60:29:3e	109	Request, Identity
295	0.024719	MurataMa_60:29:3e	Cisco_23:c6:4c	115	Response, Identity
299	0.088306	Cisco_23:c6:4c	MurataMa_60:29:3e	110	Request, Protected EAP (EAP-PEAP)
301	0.003845	MurataMa_60:29:3e	Cisco_23:c6:4c	241	Client Hello
308	0.000000	Cisco_23:c6:4c	MurataMa_60:29:3e	101	Request, Protected EAP (EAP-PEAP)
310	0.003464	MurataMa_60:29:3e	Cisco_23:c6:4c	110	Response, Protected EAP (EAP-PEAP)
312	0.003413	Cisco_23:c6:4c	MurataMa_60:29:3e	1321	Server Hello, Certificate, Server Key Exchange, Server Hello Done
314	0.010654	MurataMa_60:29:3e	Cisco_23:c6:4c	236	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
316	0.002056	Cisco_23:c6:4c	MurataMa_60:29:3e	165	Change Cipher Spec, Encrypted Handshake Message
318	0.002469	MurataMa_60:29:3e	Cisco_23:c6:4c	110	Response, Protected EAP (EAP-PEAP)
320	0.002267	Cisco_23:c6:4c	MurataMa_60:29:3e	144	Application Data
322	0.002323	MurataMa_60:29:3e	Cisco_23:c6:4c	150	Application Data
324	0.001642	Cisco_23:c6:4c	MurataMa_60:29:3e	144	Application Data
326	0.003633	MurataMa_60:29:3e	Cisco_23:c6:4c	152	Application Data
328	0.001511	Cisco_23:c6:4c	MurataMa_60:29:3e	150	Application Data
330	0.002657	MurataMa_60:29:3e	Cisco_23:c6:4c	150	Application Data
332	0.001764	Cisco_23:c6:4c	MurataMa_60:29:3e	108	Success
334	0.000000	Cisco_23:c6:4c	MurataMa_60:29:3e	221	Key (Message 1 of 4)
336	0.020090	MurataMa_60:29:3e	Cisco_23:c6:4c	342	Key (Message 2 of 4)
338	0.002610	Cisco_23:c6:4c	MurataMa_60:29:3e	391	Key (Message 3 of 4)
341	0.000000	MurataMa_60:29:3e	Cisco_23:c6:4c	199	Key (Message 4 of 4)
342	0.000000	MurataMa_60:29:3e	Cisco_23:c6:4c	199	Key (Message 4 of 4)
367	0.046878	MurataMa_60:29:3e	Broadcast	462	QoS Data, SN=2, FN=0, Flags=.p.....TC
397	0.024374	MurataMa_60:29:3e	Broadcast	144	QoS Data, SN=3, FN=0, Flags=.p.....TC
412	0.002264	MurataMa_60:29:3e	10.6.1.12	274	QoS Data, SN=4, FN=0, Flags=.p.....TC
868	0.022004	MurataMa_60:29:3e	Cisco_9f:c3:2c	243	Authentication, SN=466, FN=0, Flags=........C
870	0.001272	Cisco_9f:c3:2c	MurataMa_60:29:3e	239	Authentication, SN=0, FN=0, Flags=........C
872	0.001386	MurataMa_60:29:3e	Cisco_9f:c3:2c	480	Reassociation Request, SN=467, FN=0, Flags=........C, SSID=11renable
875	0.002663	Cisco_9f:c3:2c	MurataMa_60:29:3e	467	Reassociation Response, SN=1, FN=0, Flags=........C
899	0.012668	MurataMa_60:29:3e	Cisco_b1:6c:fd	144	QoS Data, SN=1, FN=0, Flags=.p.....TC
901	0.007196	MurataMa_60:29:3e	Broadcast	144	QoS Data, SN=2, FN=0, Flags=.p.....TC

Figure 6-12 802.11r (FT) initial association and roam (over-the-air)

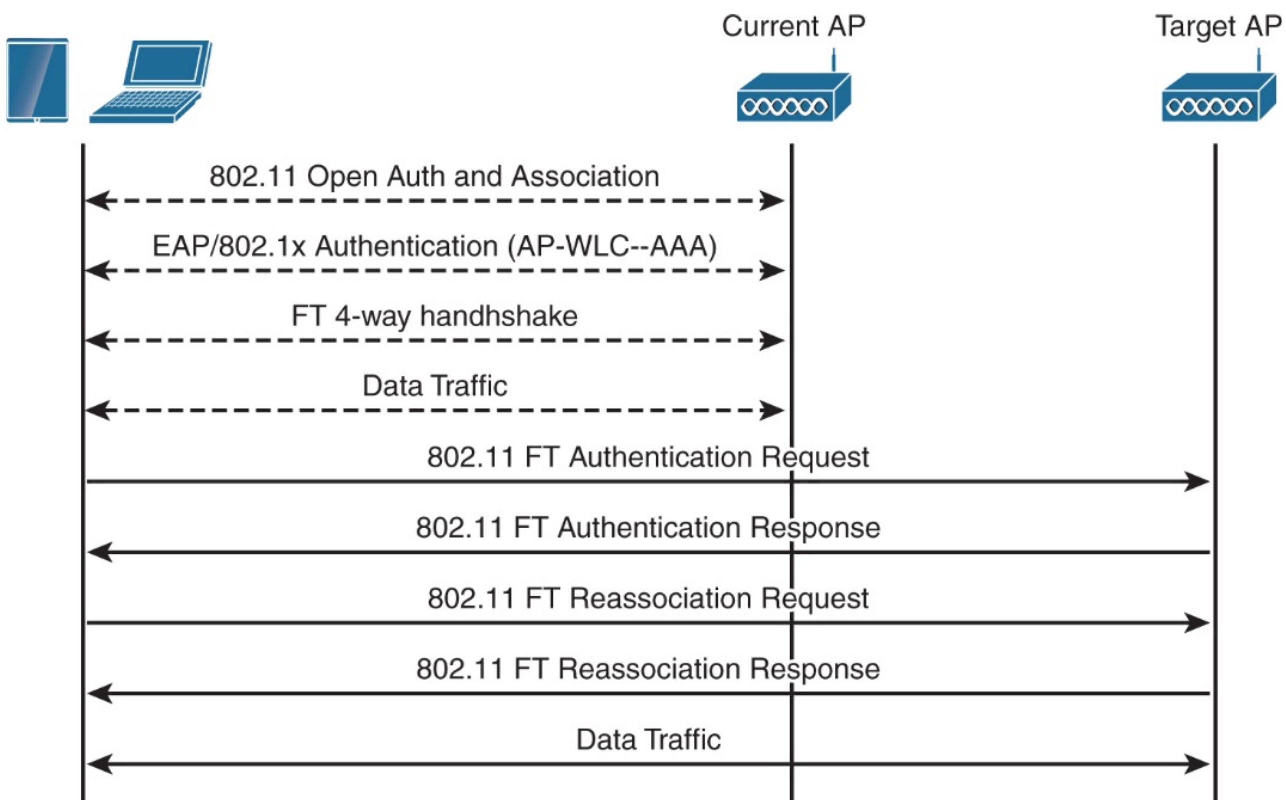


Figure 6-13 FT roam over-the-air

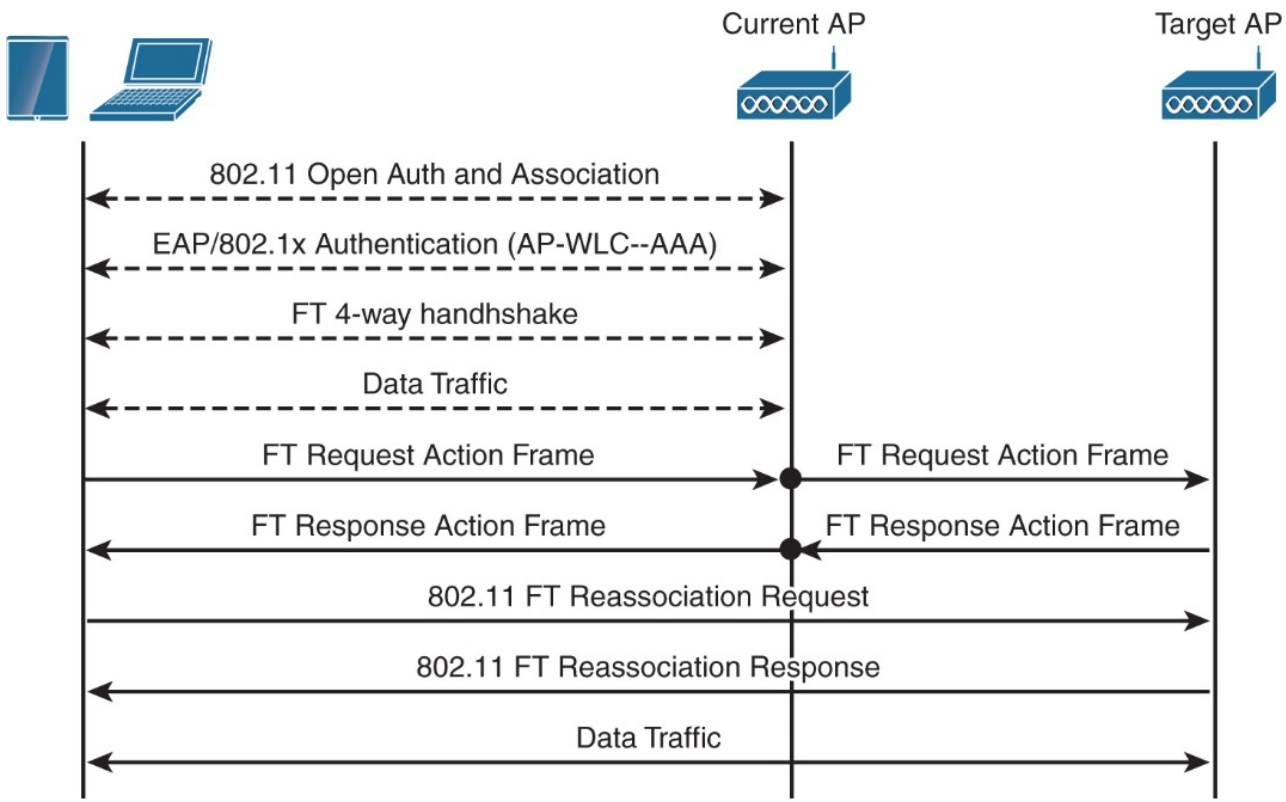


Figure 6-14 FT roam over-the-DS

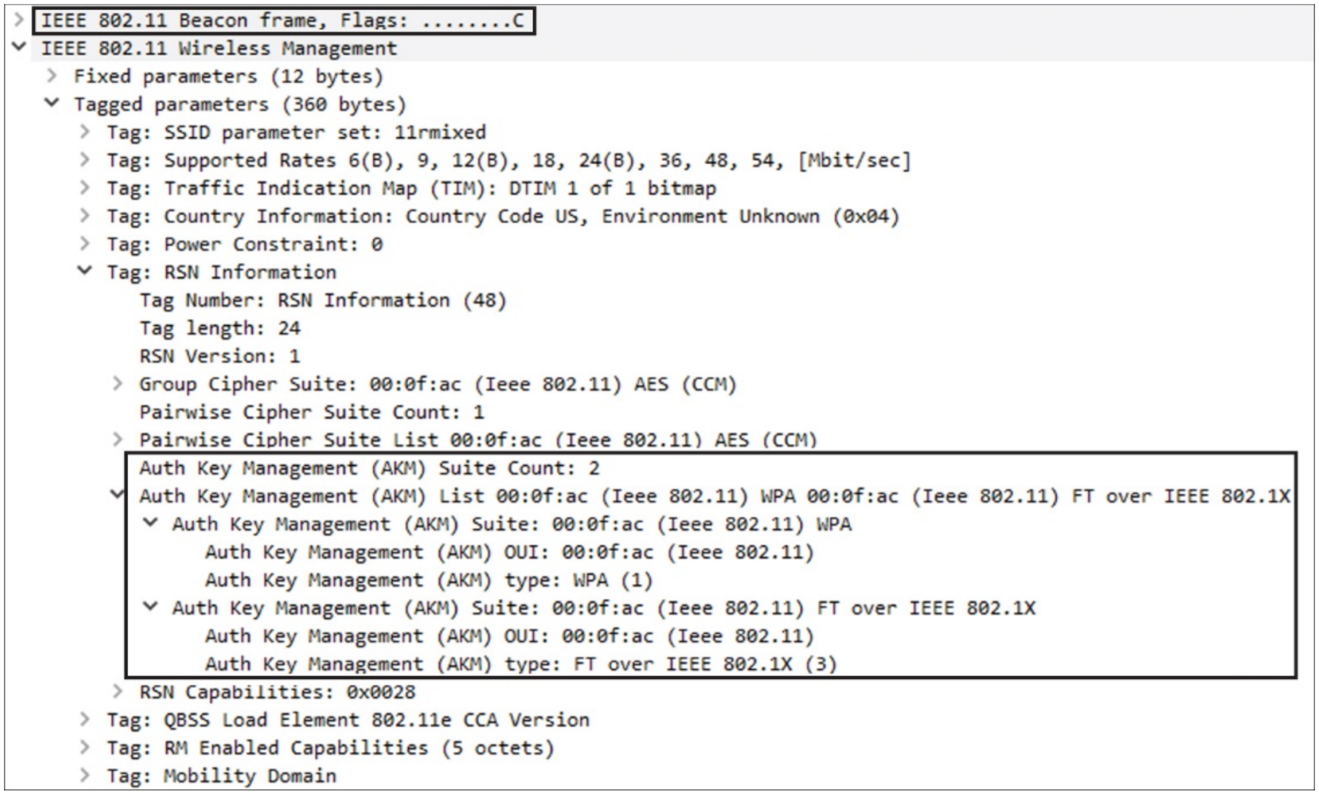


Figure 6-15 AKM in RSN IE in beacon of mixed-mode FT WLAN

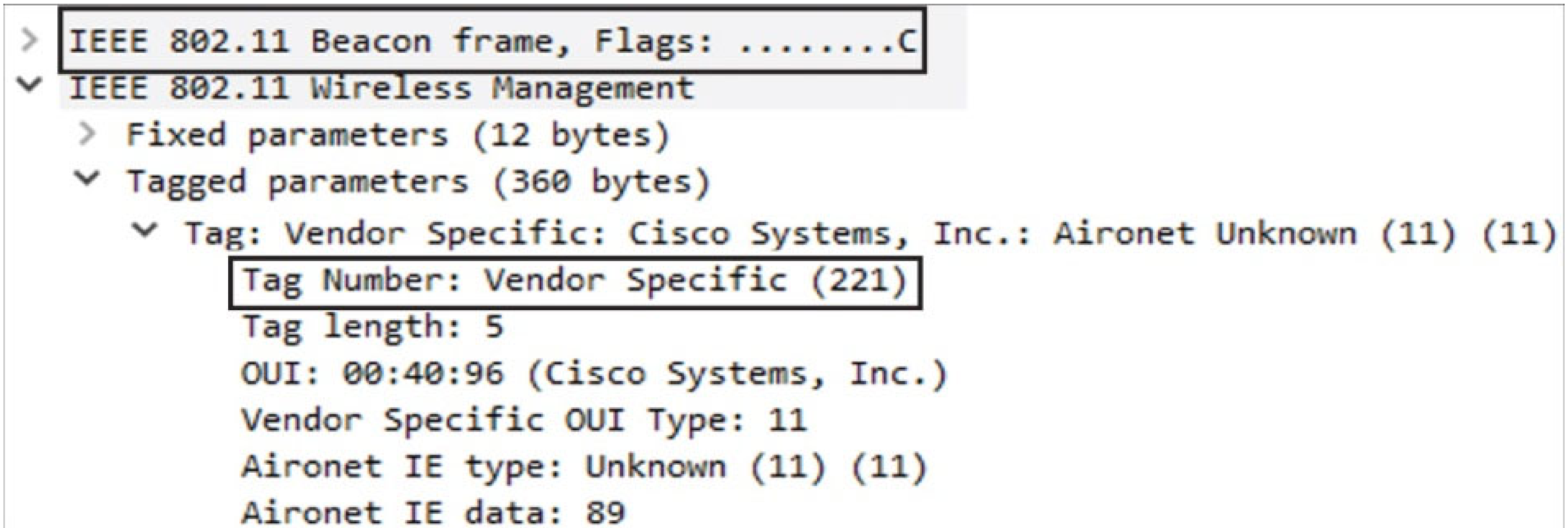


Figure 6-16 Aironet IE in beacon of mixed-mode FT WLAN

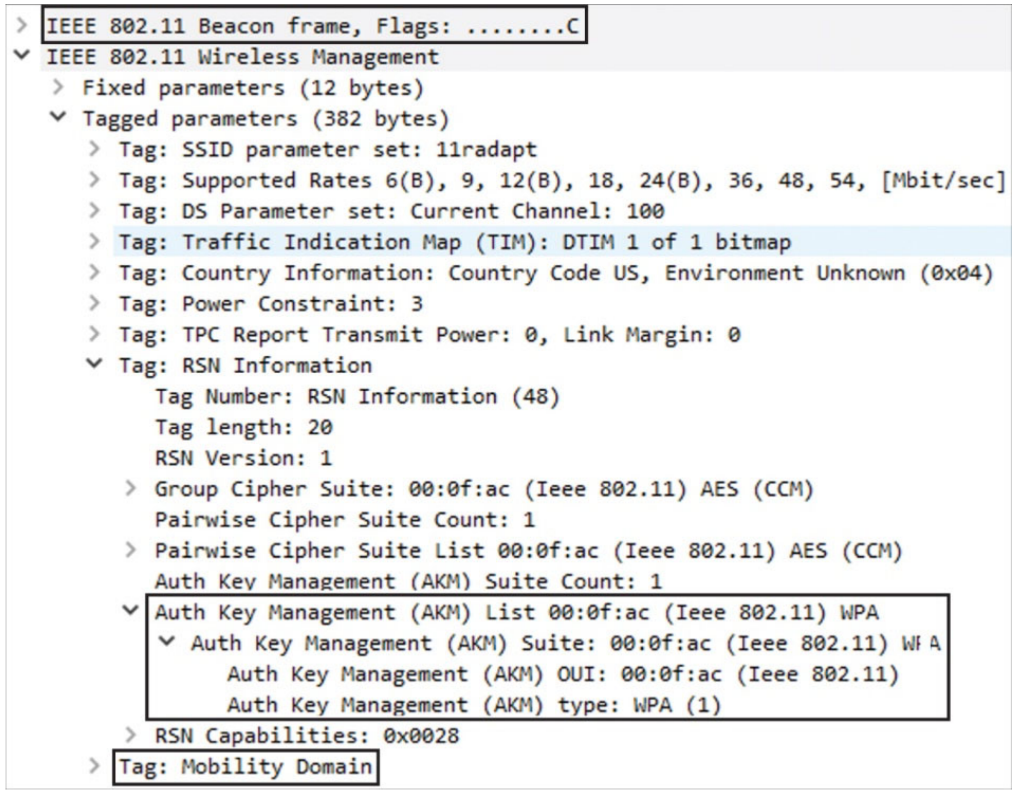


Figure 6-17 AKM in RSN IE of beacon of adaptive FT WLAN

```
> IEEE 802.11 Beacon frame, Flags: ........C
v IEEE 802.11 Wireless Management
    > Fixed parameters (12 bytes)
    v Tagged parameters (382 bytes)
        > Tag: SSID parameter set: 11radapt
        > Tag: Supported Rates 6(B), 9, 12(B), 18, 24(B), 36, 48, 54, [Mbit/sec]
        v Tag: Vendor Specific: Cisco Systems, Inc.: Aironet Unknown (11) (11)
              Tag Number: Vendor Specific (221)
              Tag length: 5
              OUI: 00:40:96 (Cisco Systems, Inc.)
              Vendor Specific OUI Type: 11
              Aironet IE type: Unknown (11) (11)
              Aironet IE data: c9
```

Figure 6-18 Aironet IE in beacon of adaptive FT WLAN

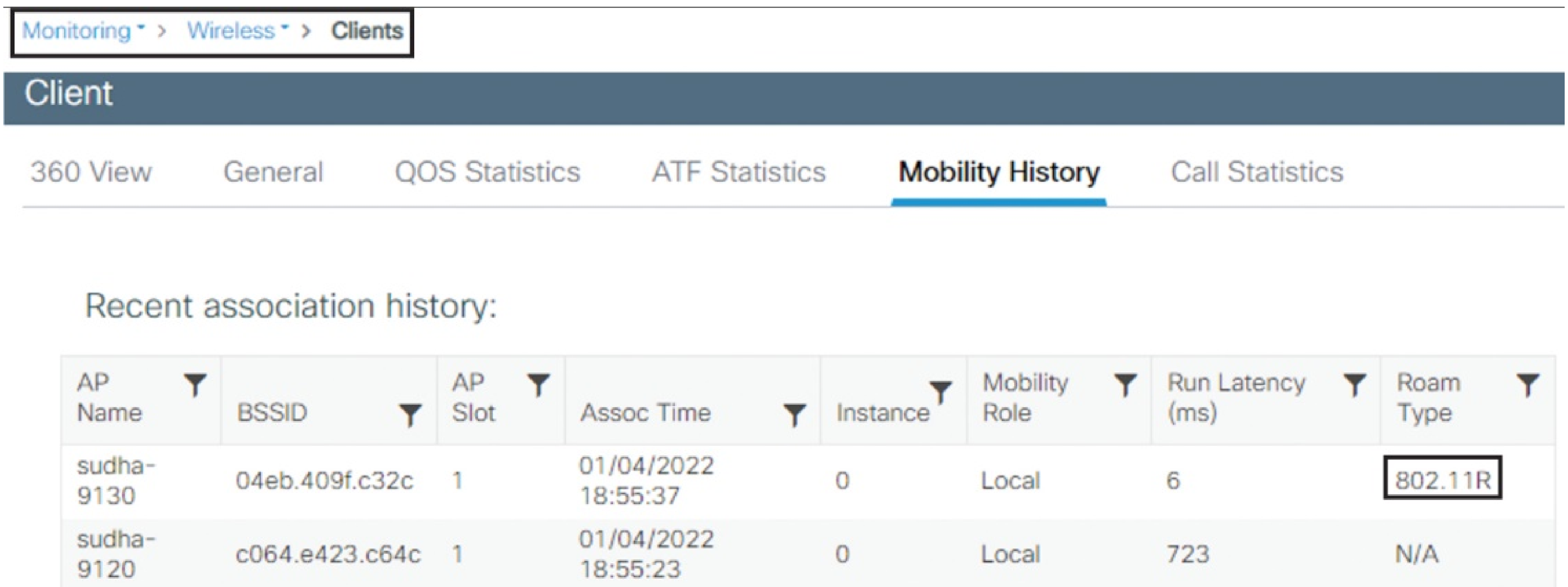


Figure 6-19 Monitoring the mobility history of a client on the C9800 GUI

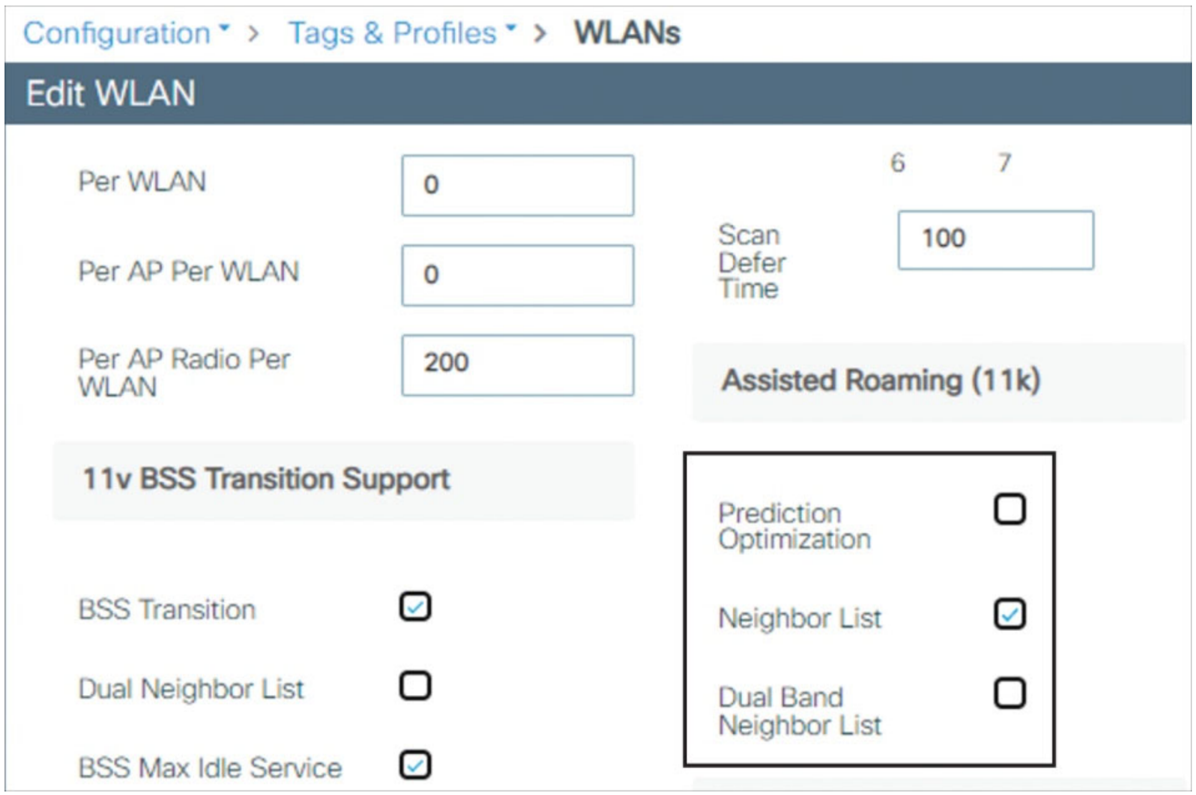


Figure 6-20 Configuring an 802.11k on the C9800 WLAN Profile

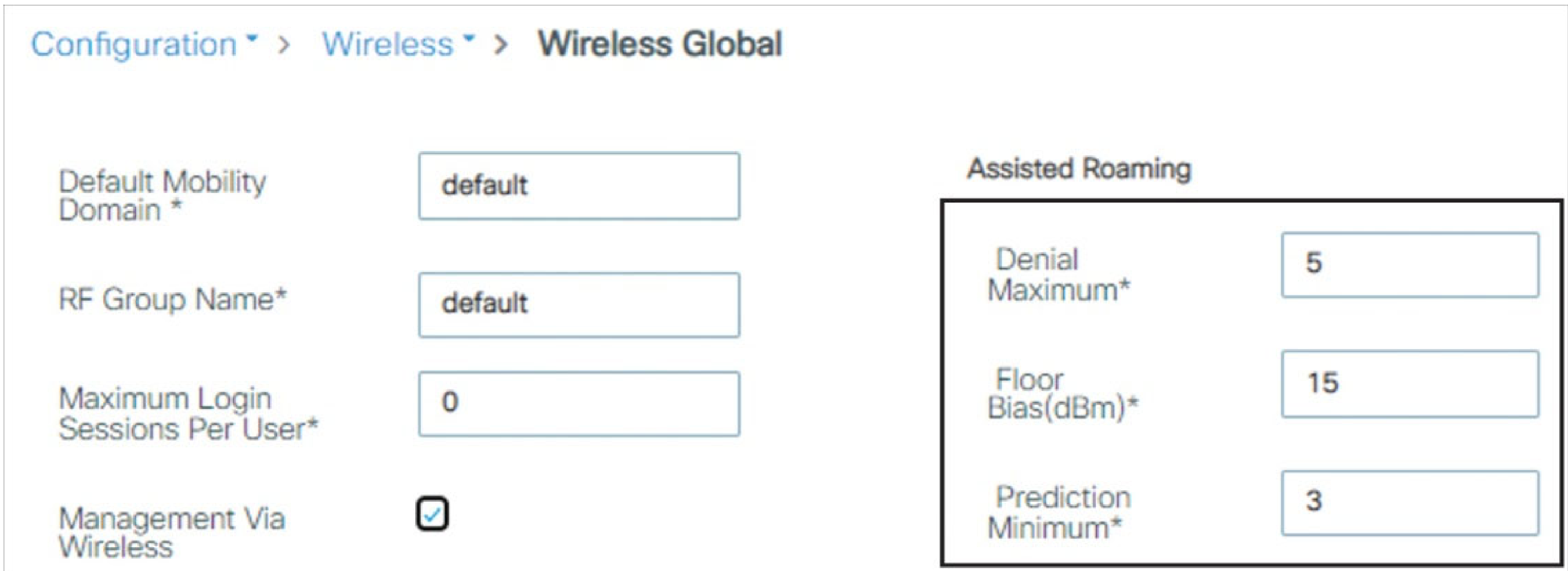


Figure 6-21 Configuring assisted roaming parameters on the C9800

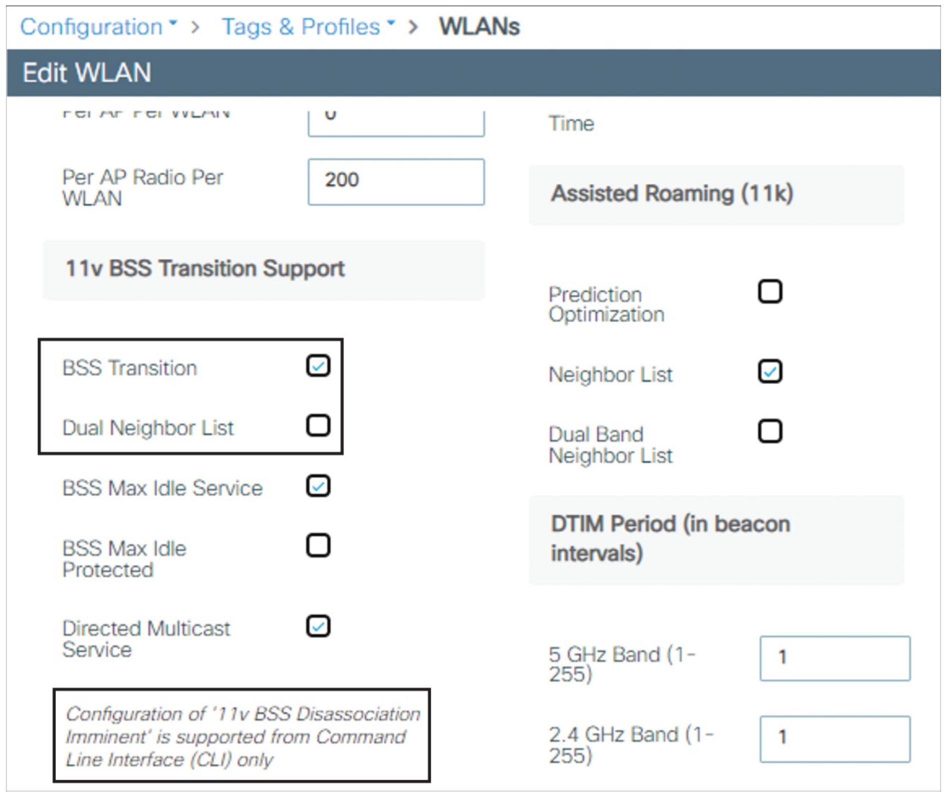


Figure 6-22 Configuring an 802.11v BSS transition on the C9800

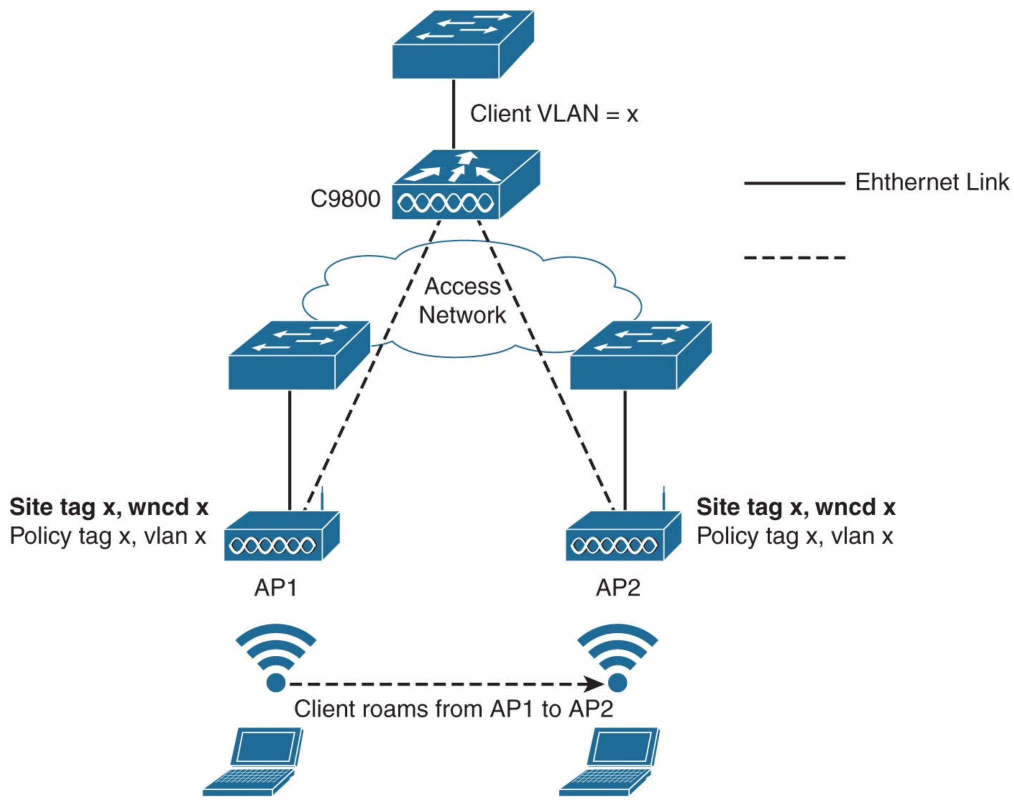


Figure 6-23 Intra-controller Intra-WNCd roaming on the C9800

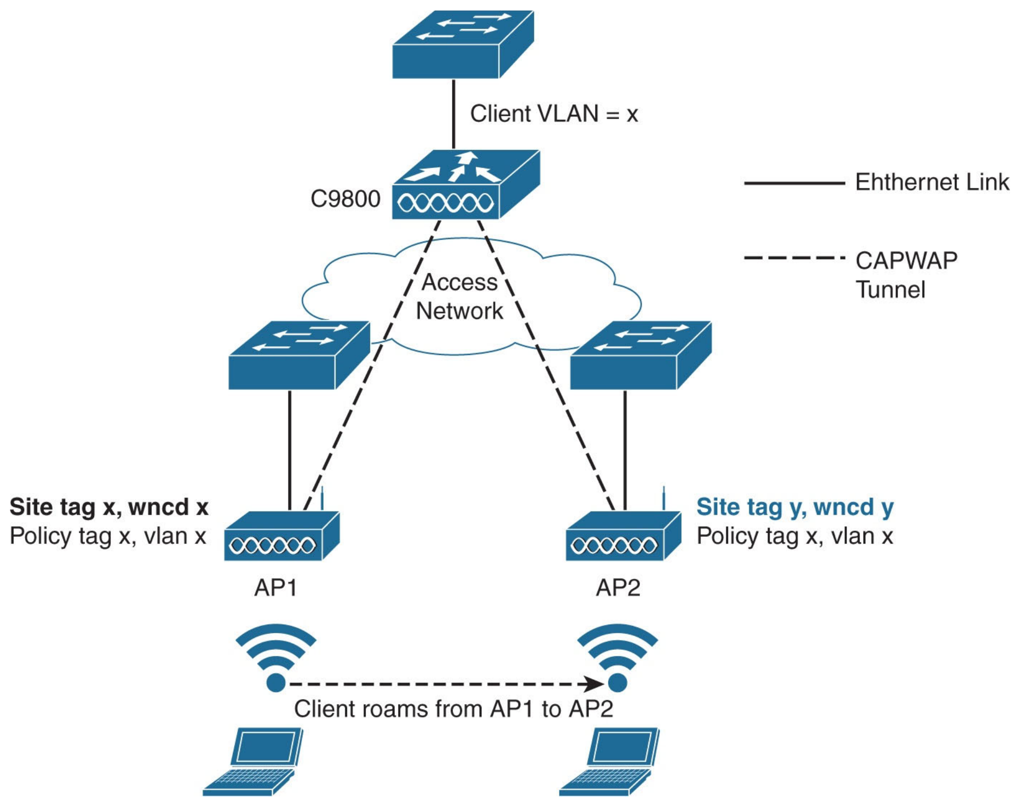


Figure 6-24 Intra-controller inter-WNCd roaming on the C9800

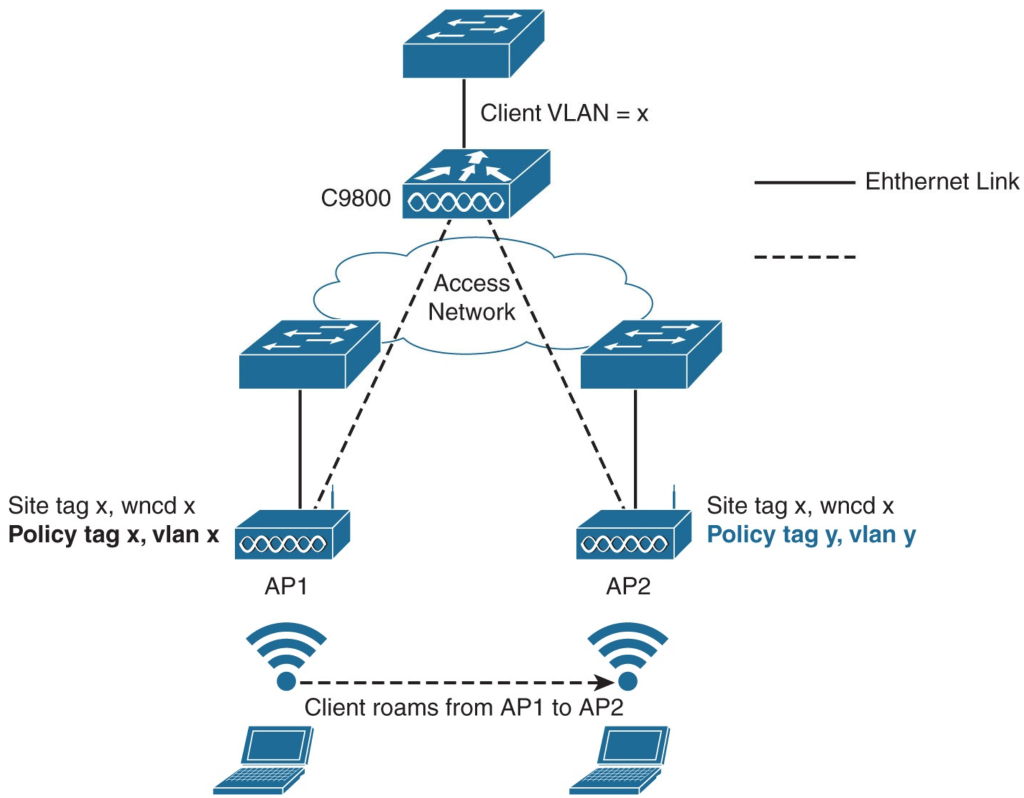


Figure 6-25 Intra-controller inter-policy profile roaming on the C9800

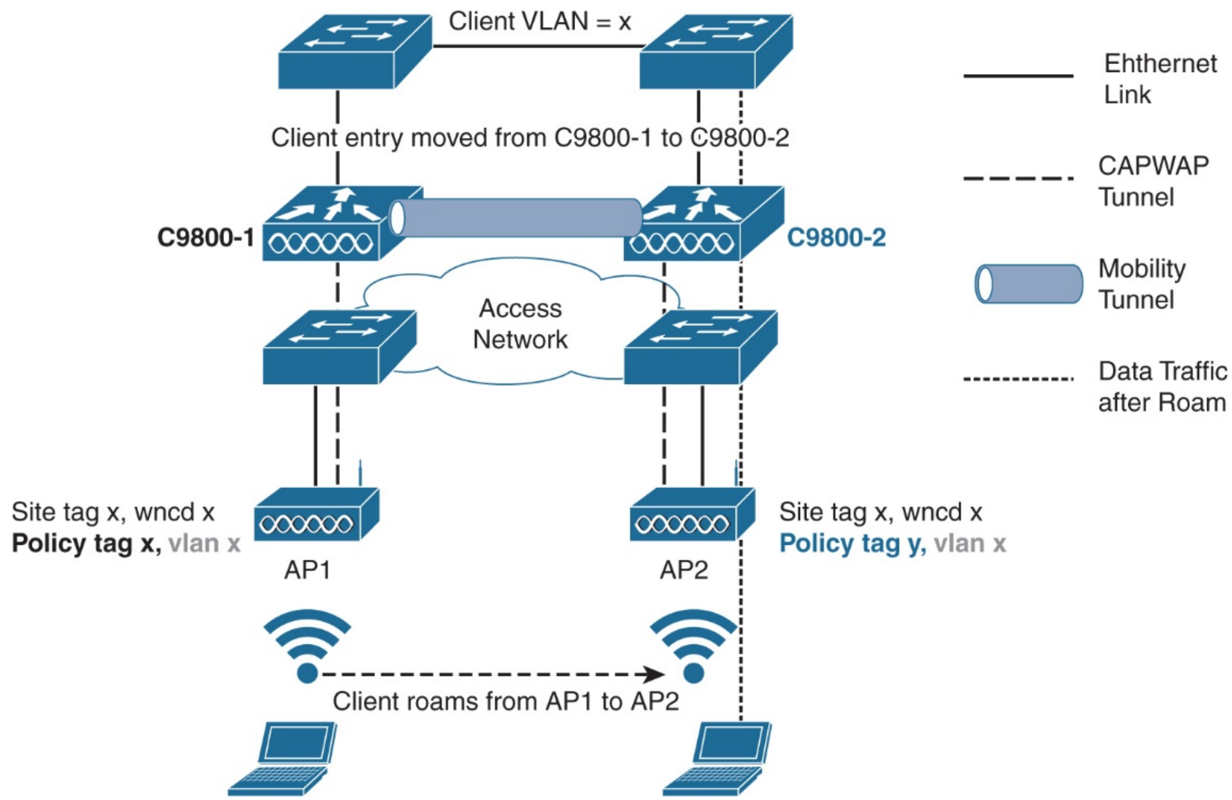


Figure 6-26 Layer 2 CAPWAP Mobility Tunnel C9800

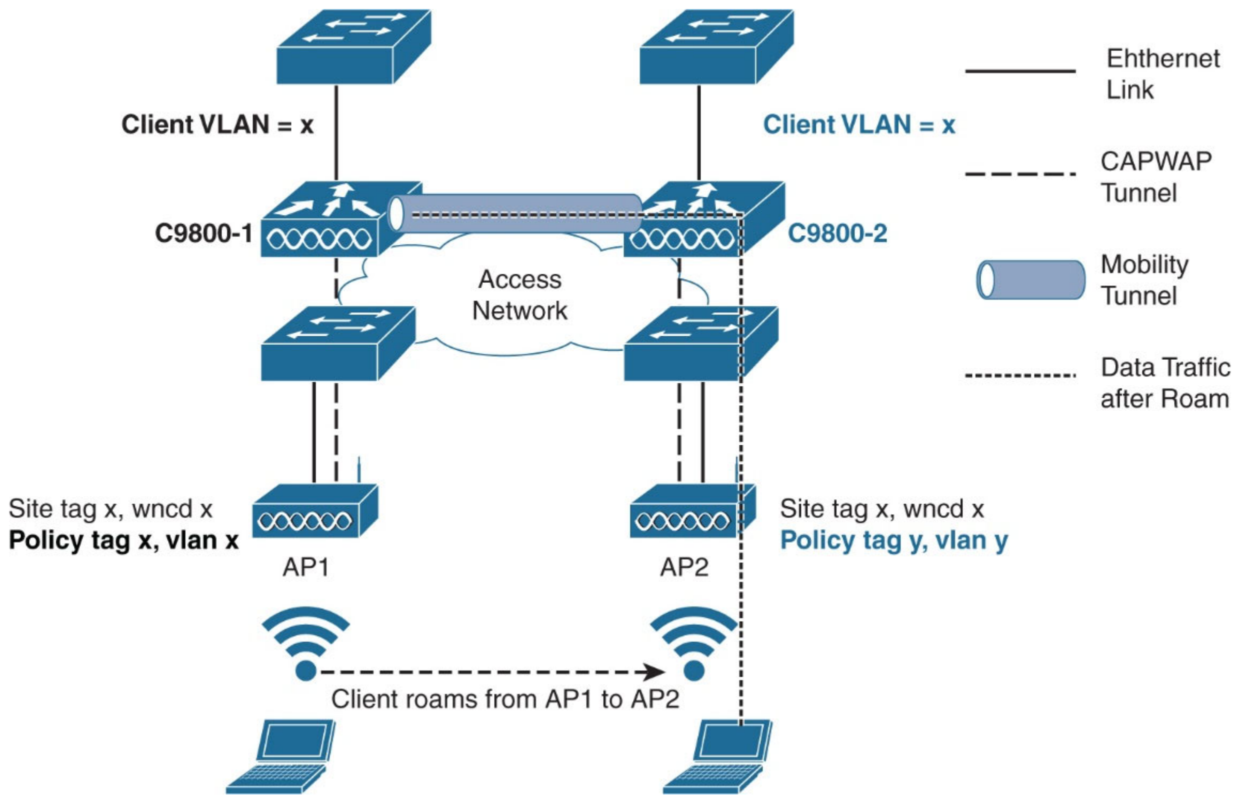


Figure 6-27 Layer 3 inter-controller roaming on the C9800

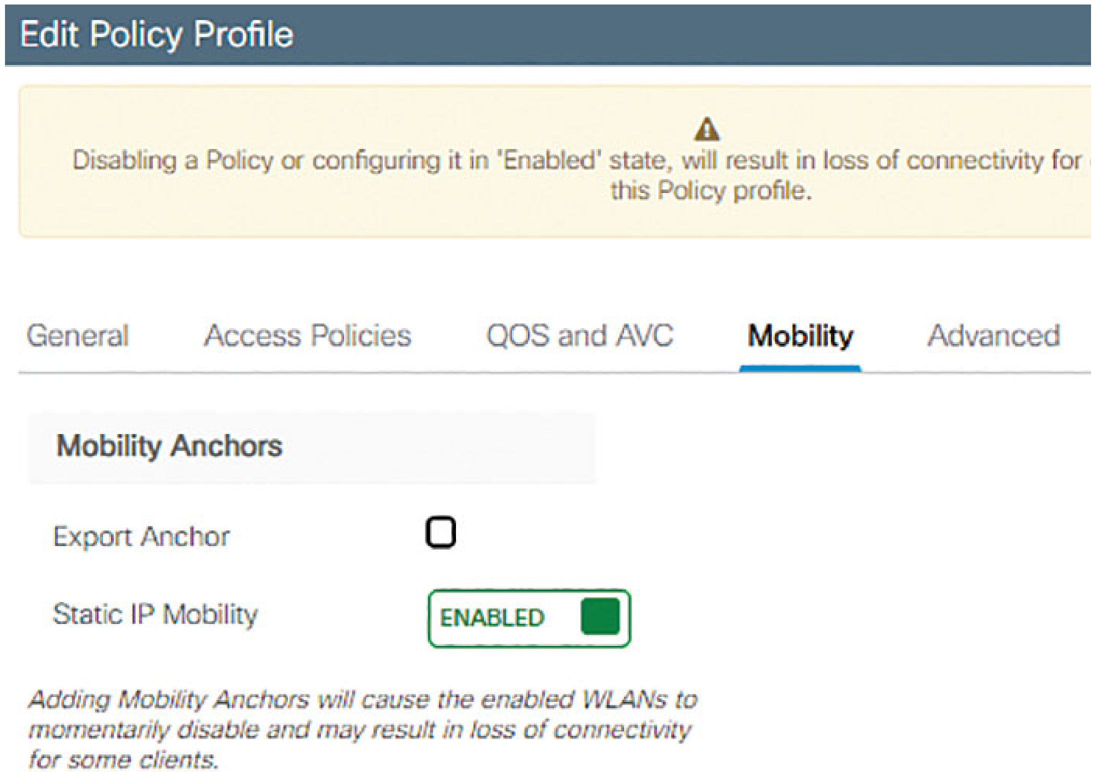


Figure 6-28 Configuring Static IP Mobility on the C9800

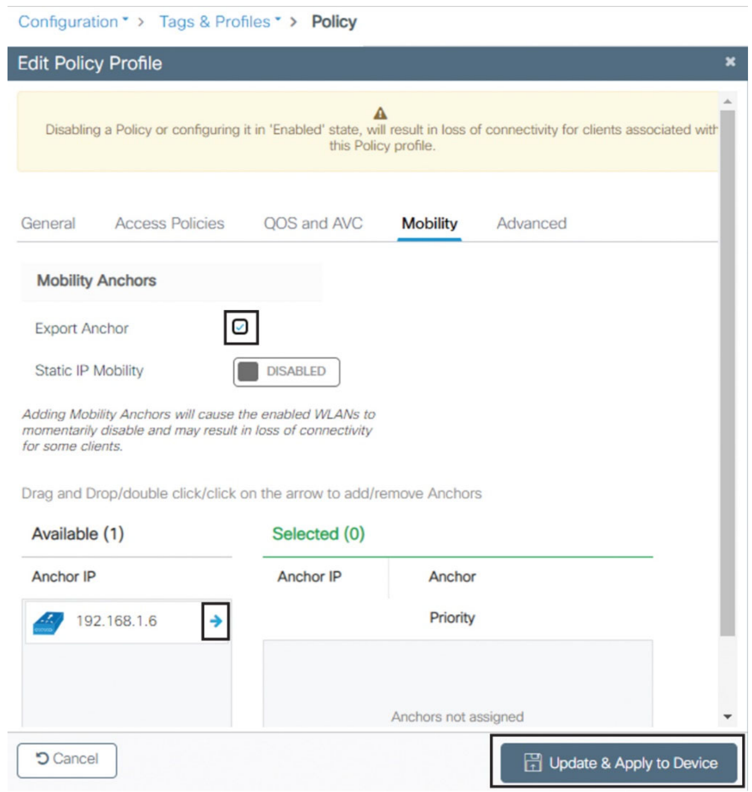


Figure 6-29 Configuring auto anchoring on the C9800

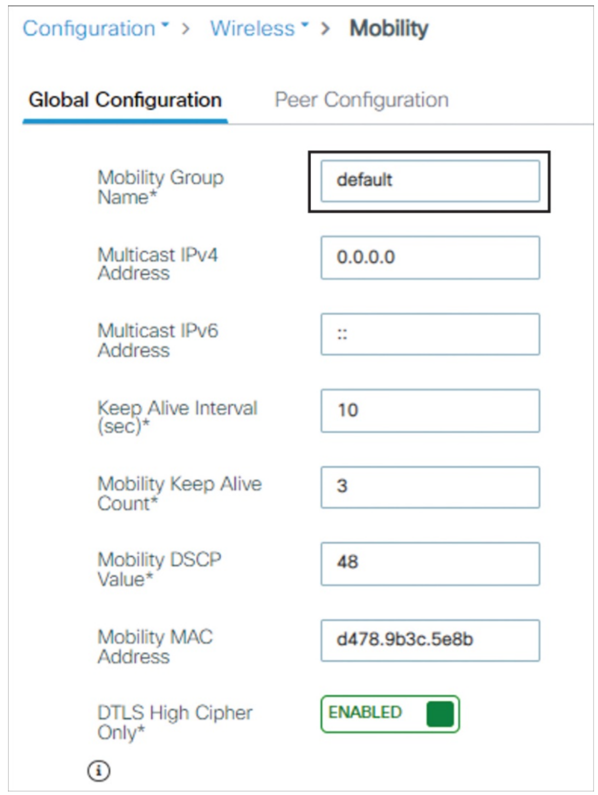


Figure 6-30 Configuring secure mobility on the C9800

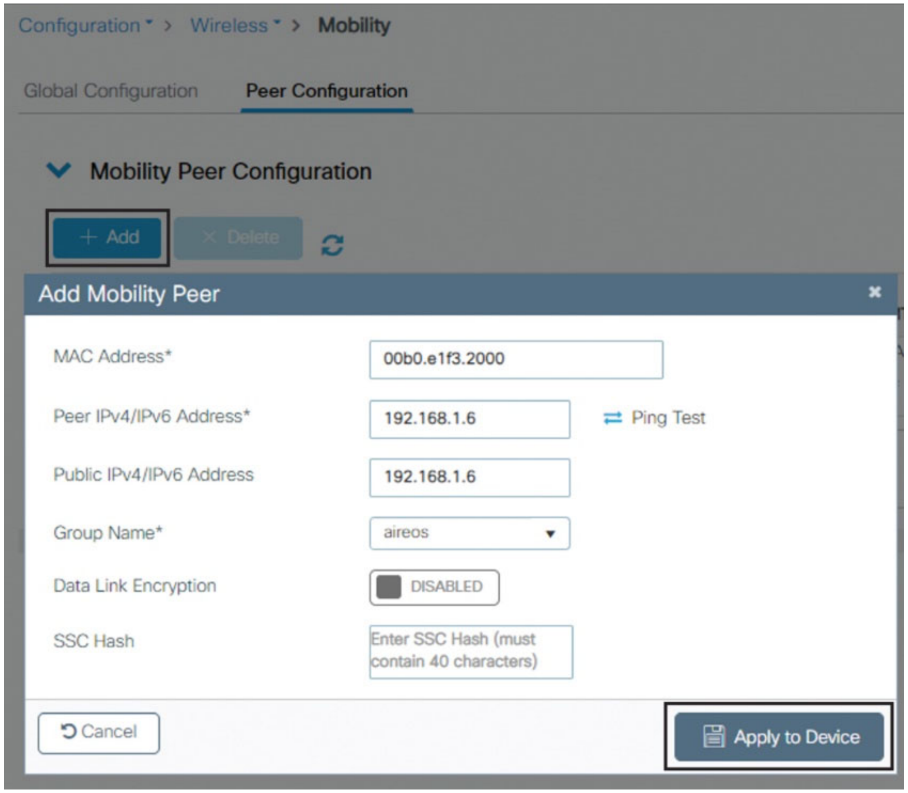


Figure 6-31 Configuring mobility peers on the C9800

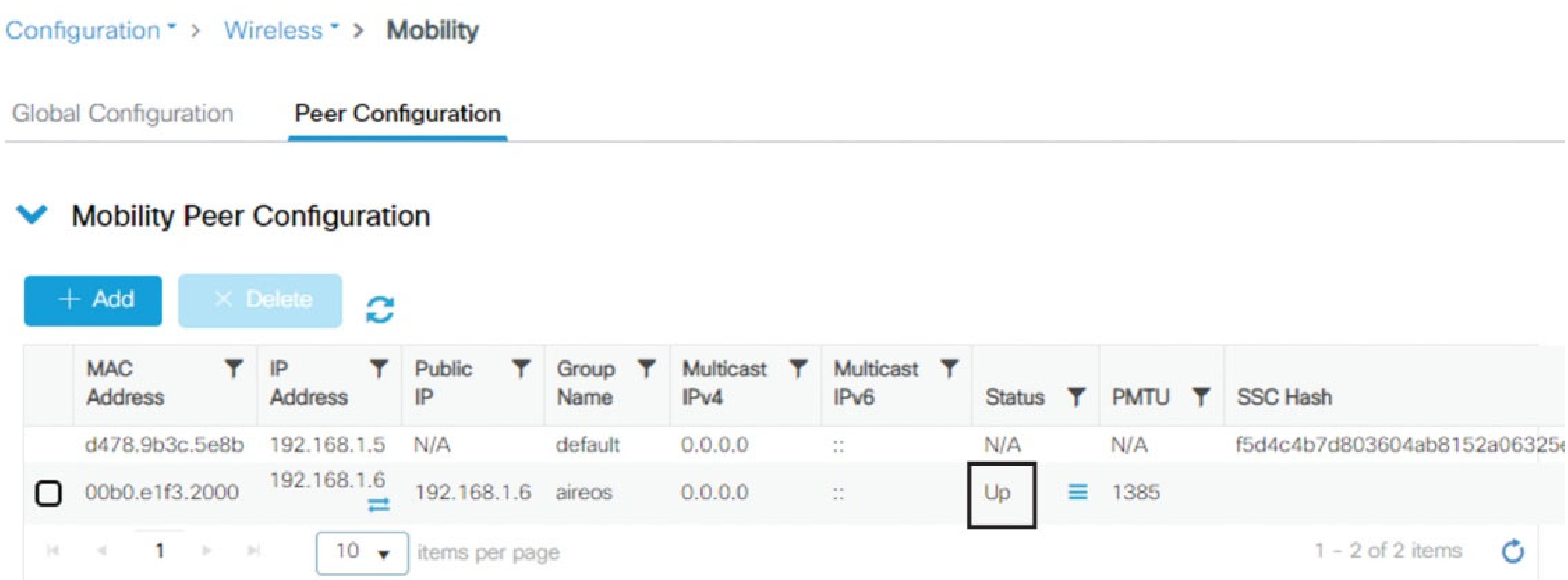


Figure 6-32 Monitoring mobility tunnel status on the C9800

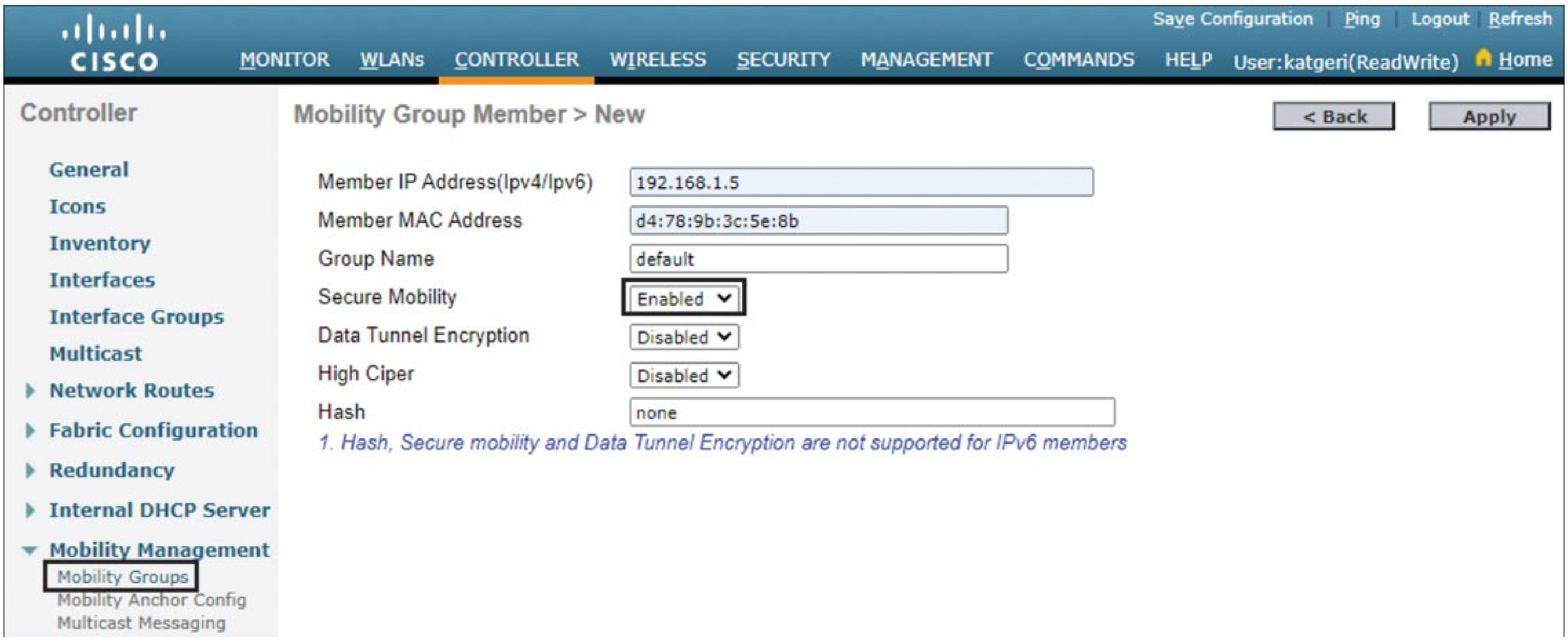


Figure 6-33 Configuring secure mobility on AireOS WLC

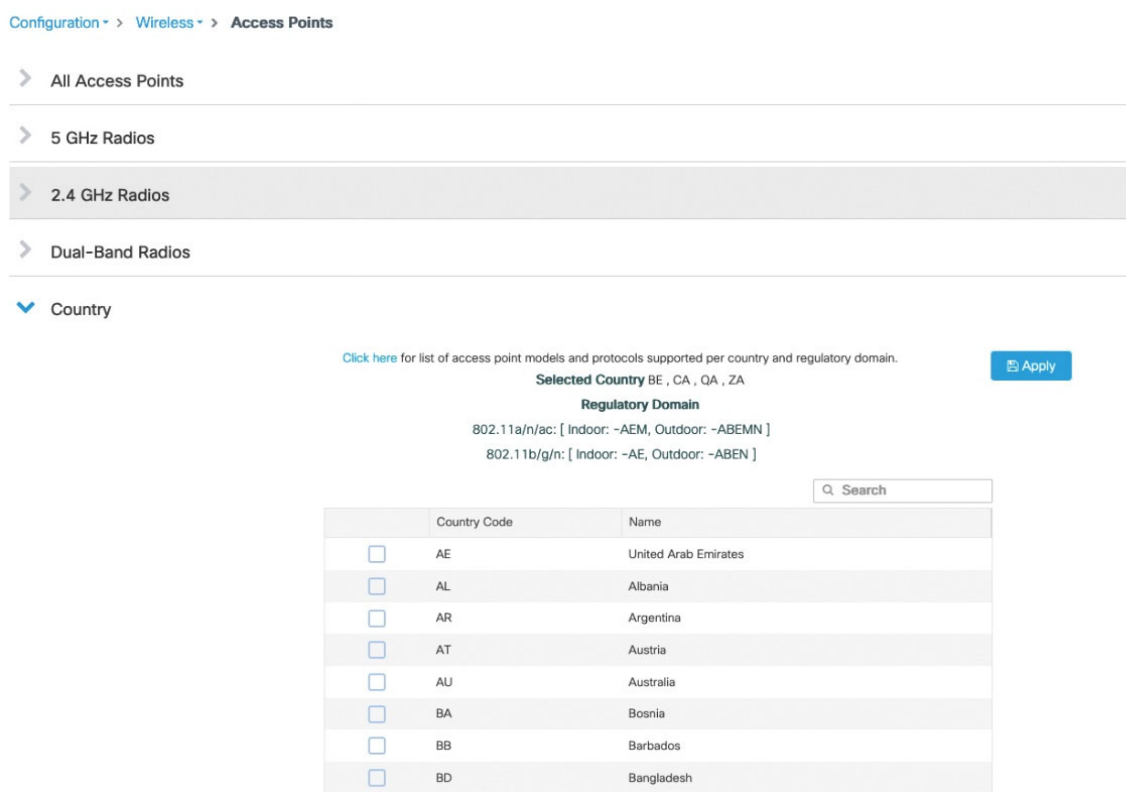


Figure 7-1 The country configuration page

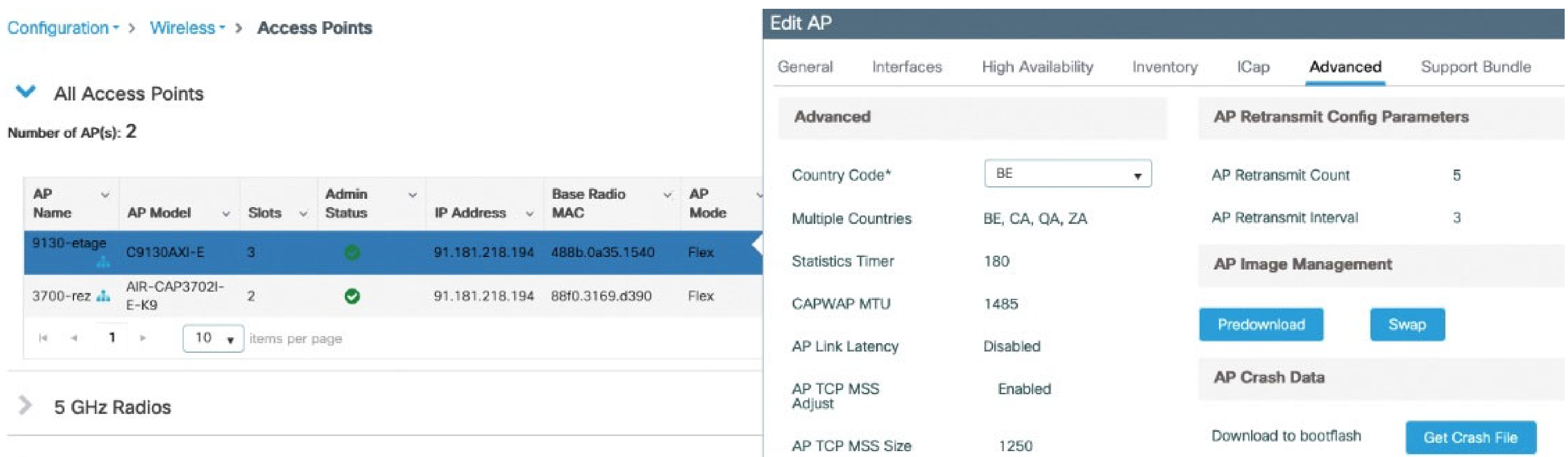


Figure 7-2 Country selection for each AP

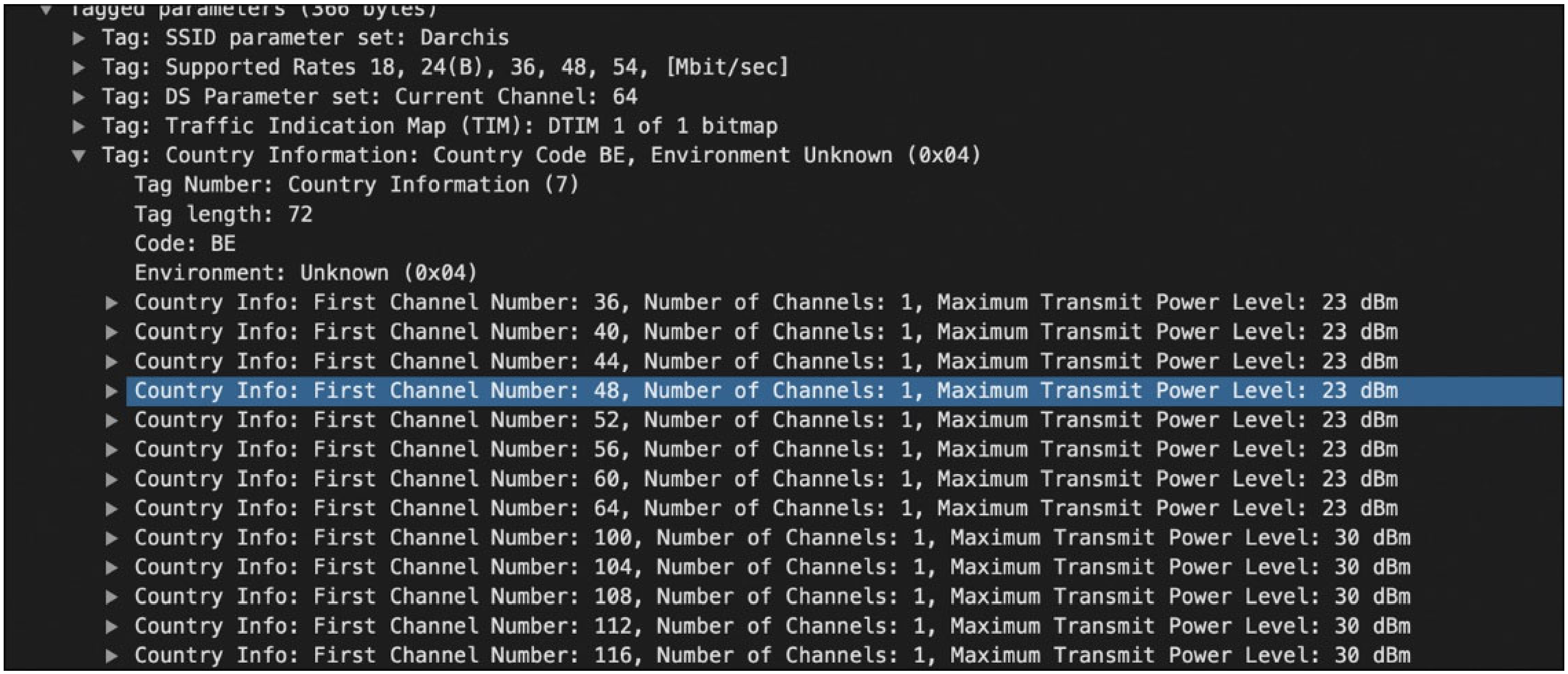


Figure 7-3 Country information element in an AP beacon

Figure 7-4 A steel-heavy industrial environment where reflections cause problems

Figure 7-5 A heavy door that can cause a lot of signal attenuation and typically breaks roaming

From *Understanding and Troubleshooting Cisco Catalyst 9800 Series Wireless Controller by* Simone Arena, Francisco Sedano Crippa, Nicolas Darchis and Sudha Katgeri(9780137492329) Cisco Press

Figure 7-6 An atrium is an open space with no attenuation connecting several floors, a Wi-Fi interference nightmare

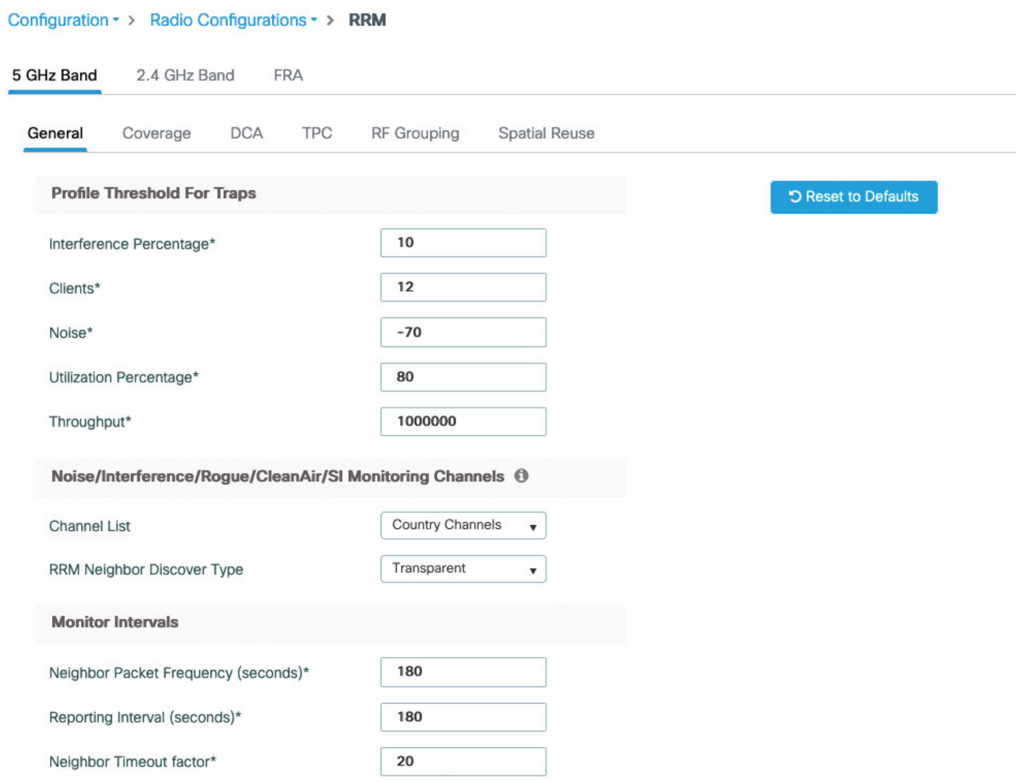


Figure 7-7 Configuring RRM data collection settings

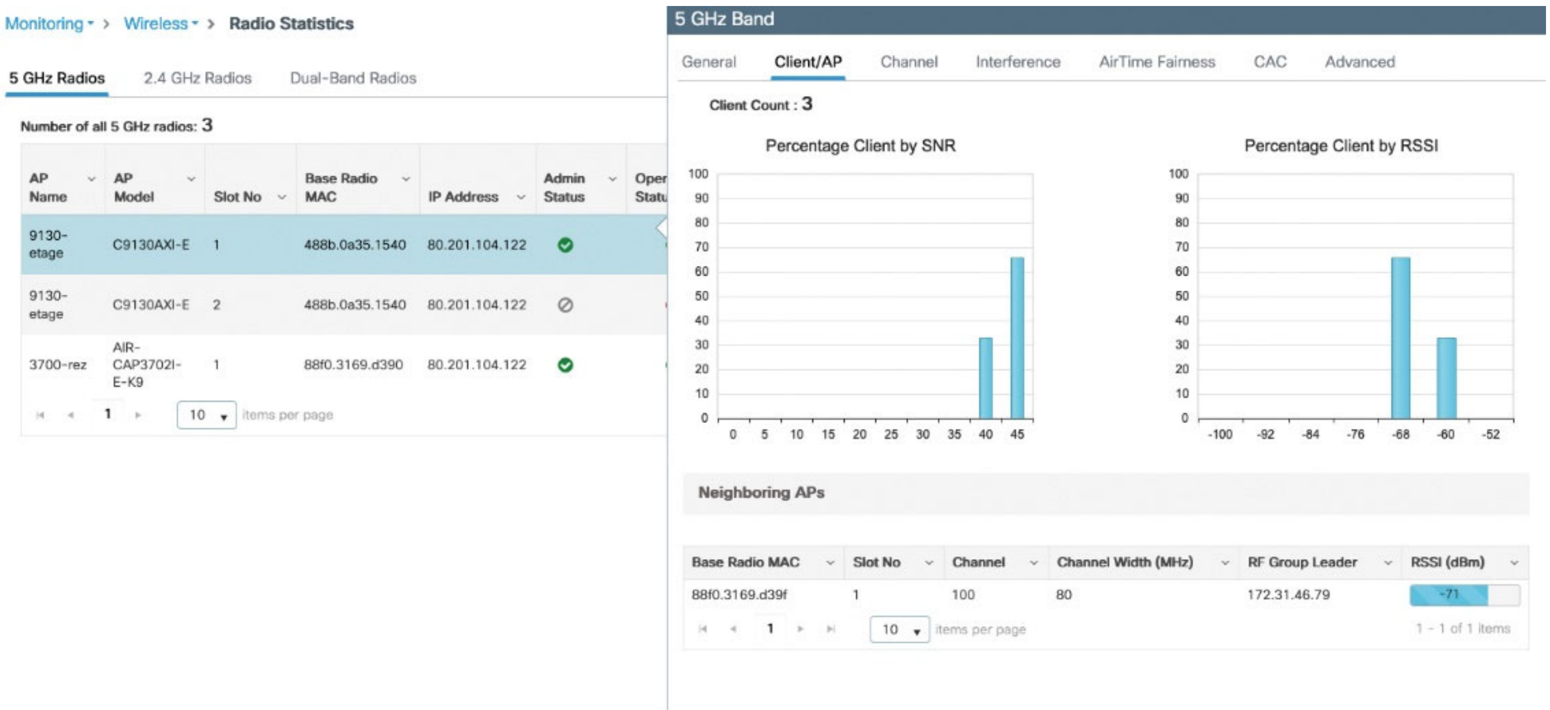


Figure 7-8 Monitoring AP neighbors

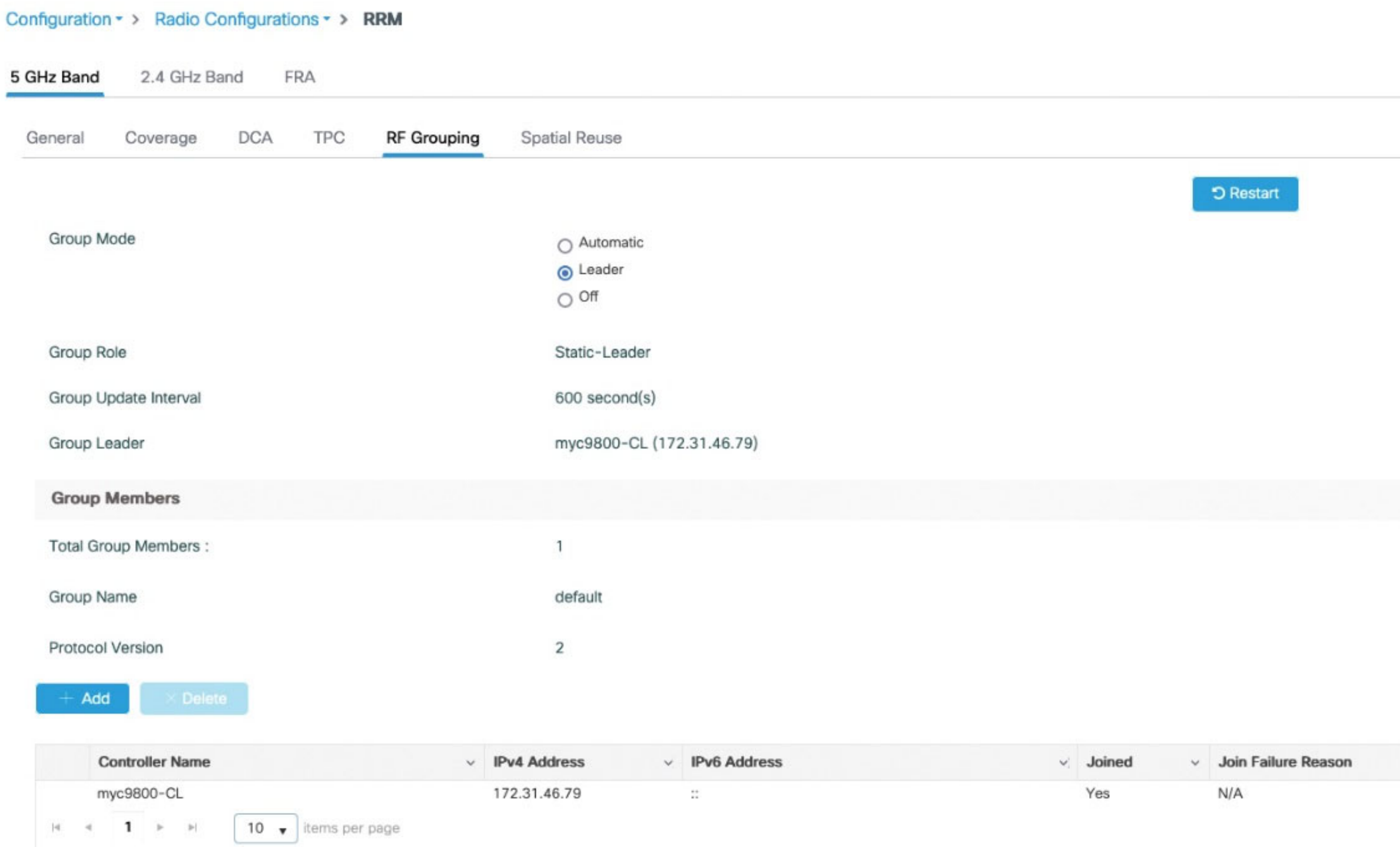


Figure 7-9 RF grouping monitoring and configuration page

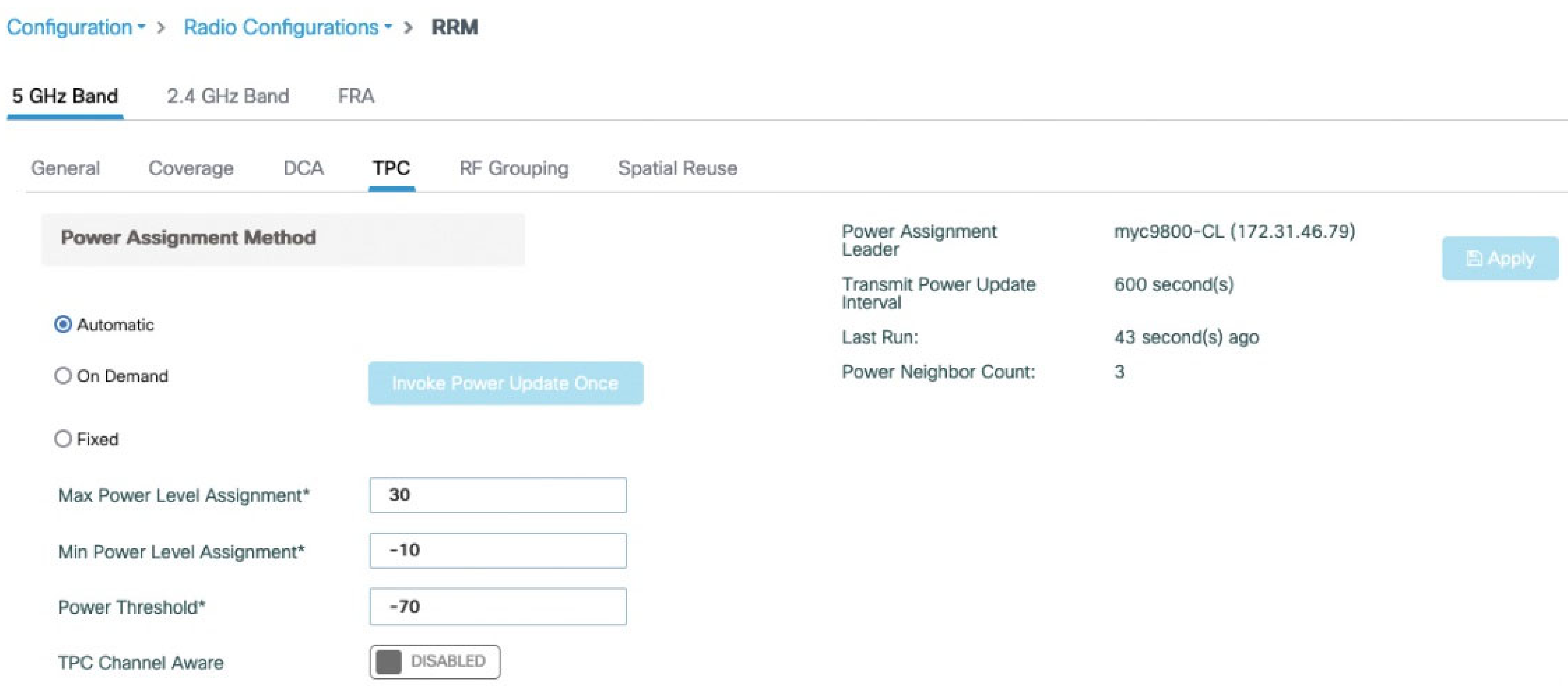


Figure 7-10 TPC settings

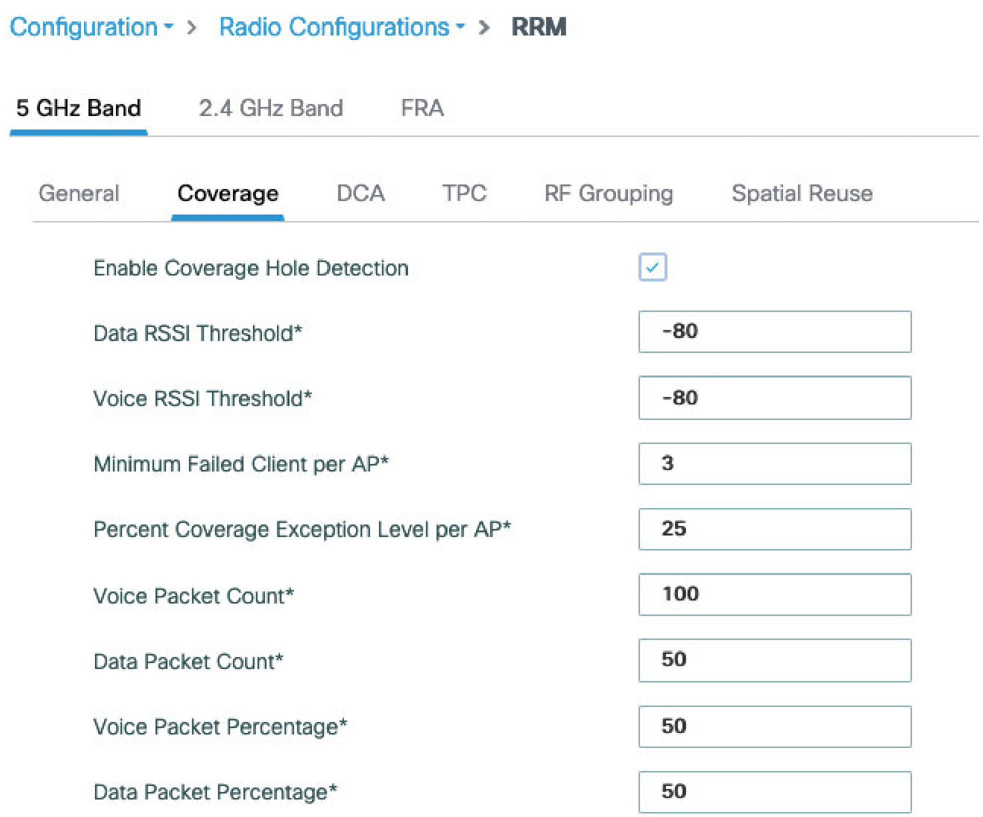


Figure 7-11 Coverage hole detection algorithm settings

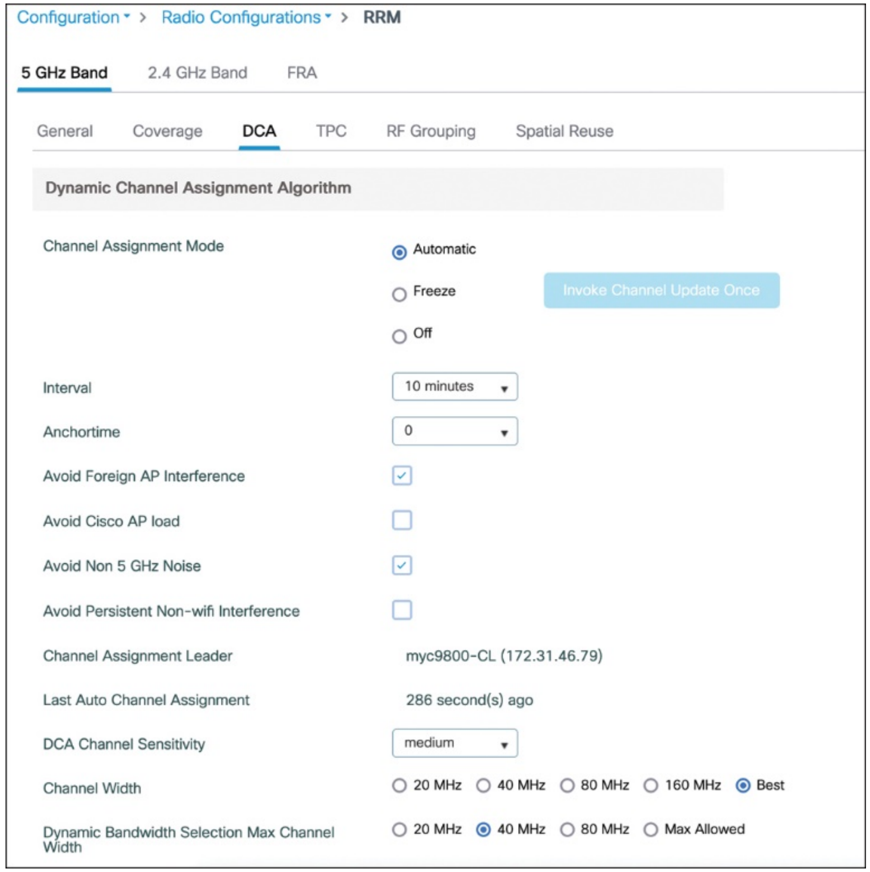


Figure 7-12 DCA settings

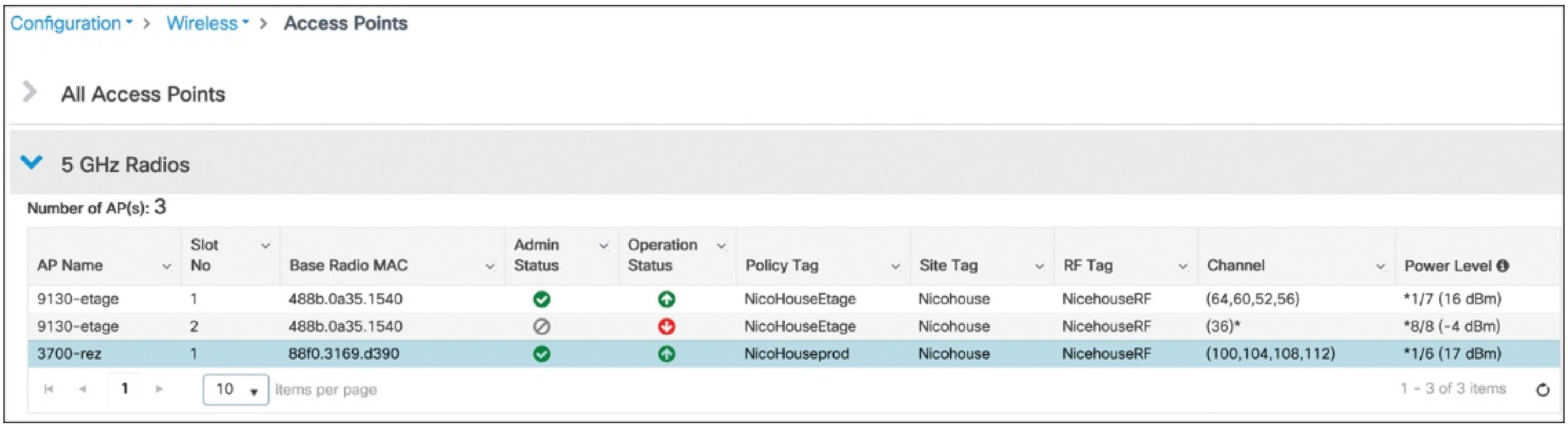


Figure 7-13 Manual channel configuration result

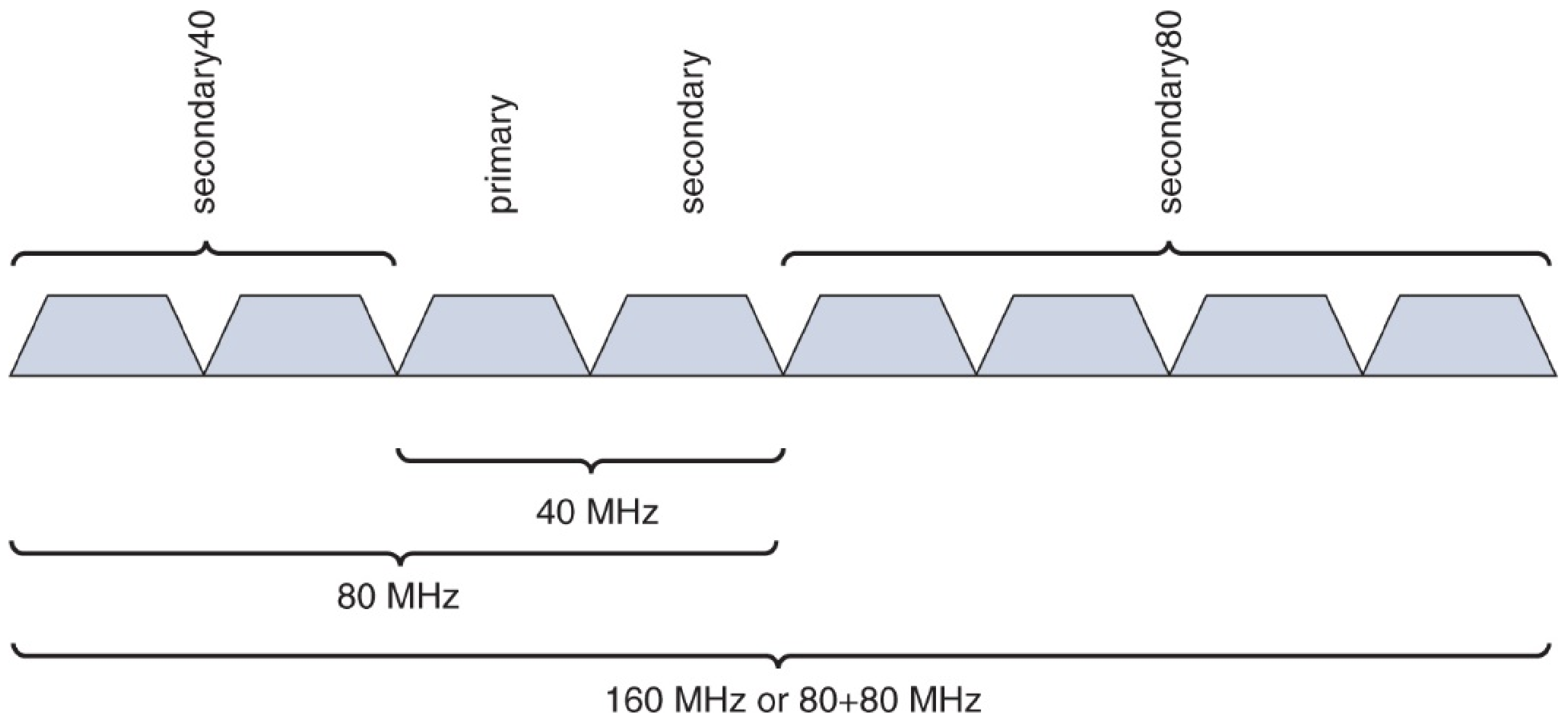


Figure 7-14 Complex primary and secondary channel plans for various channel widths

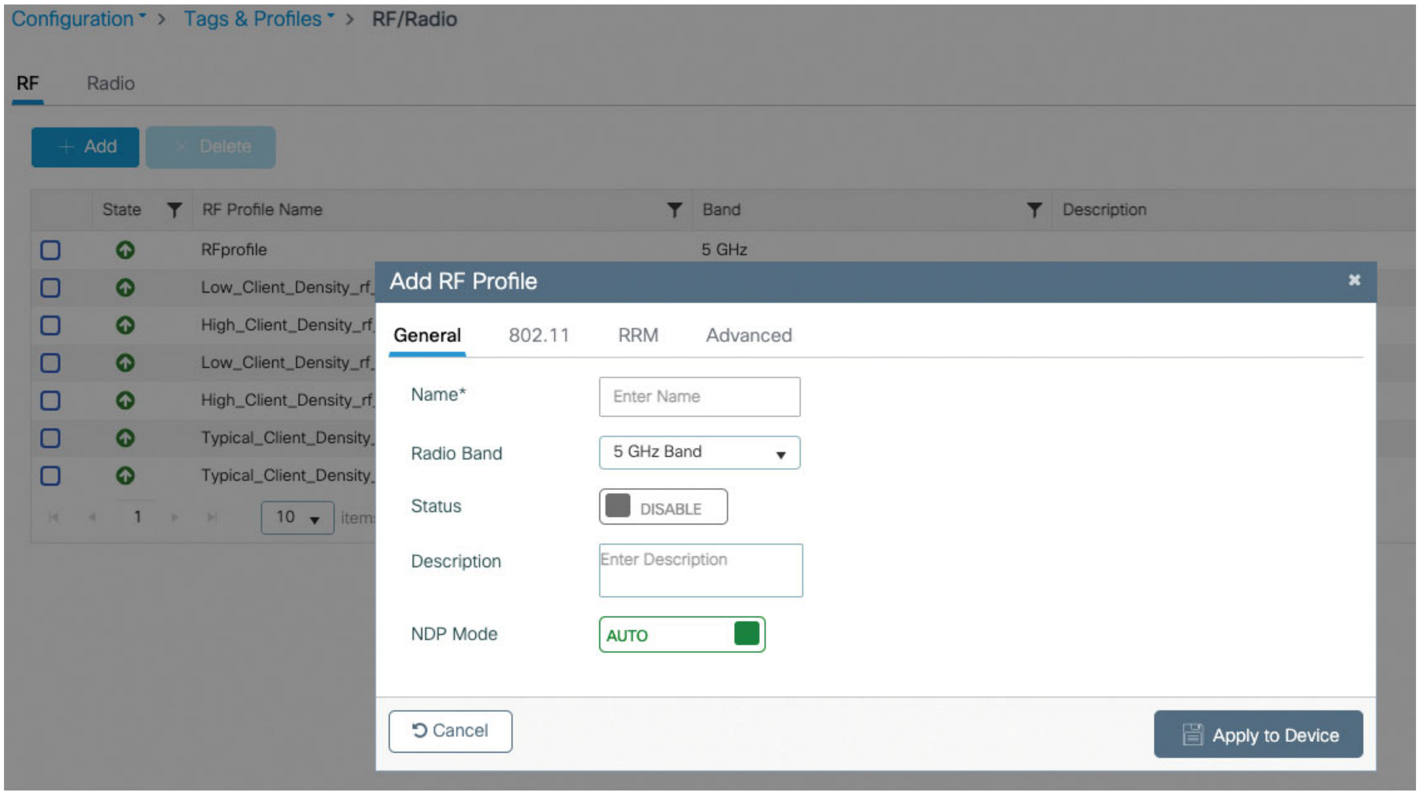


Figure 7-15 RF profile general configuration

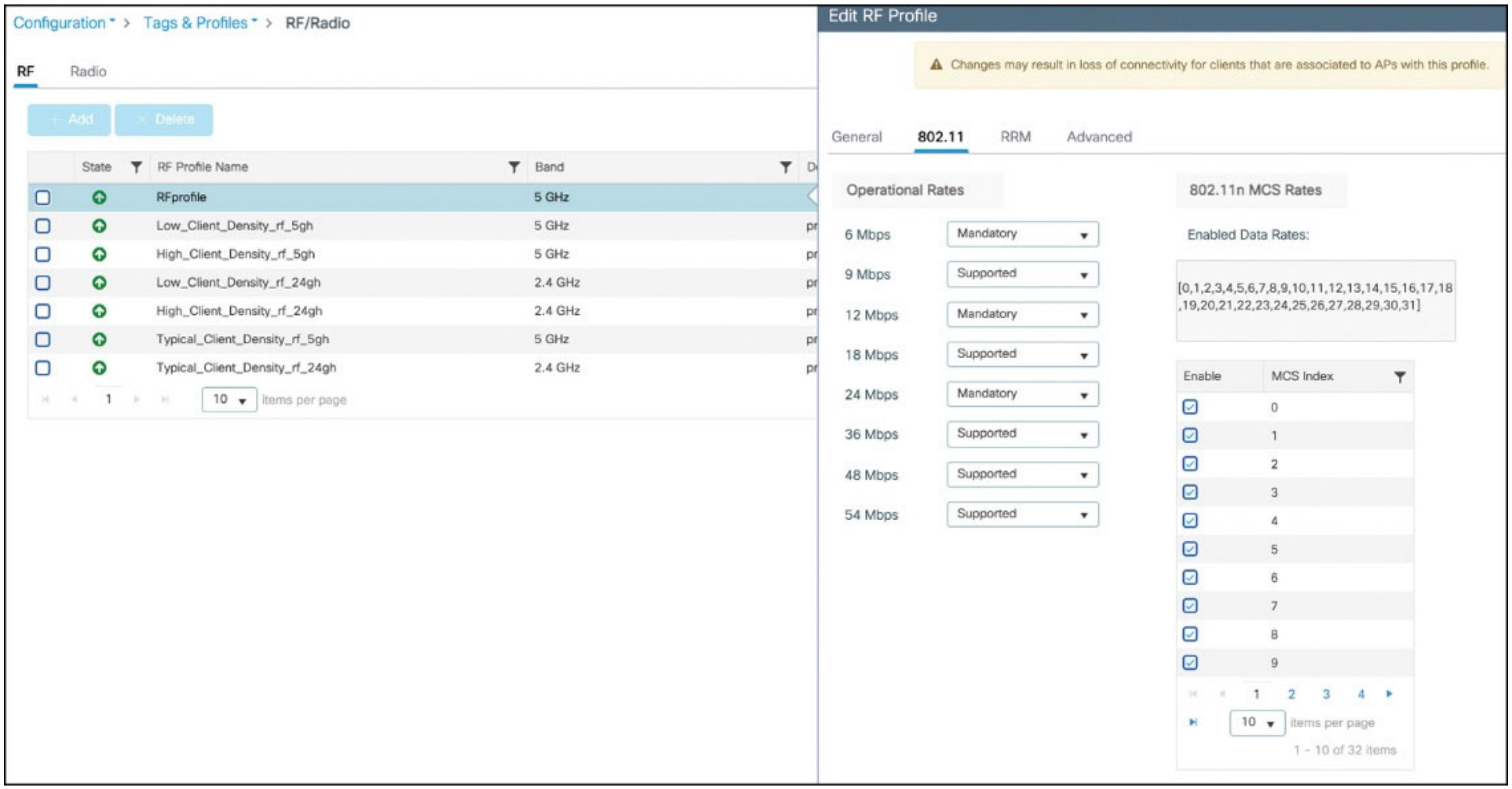


Figure 7-16 RF profile configuration

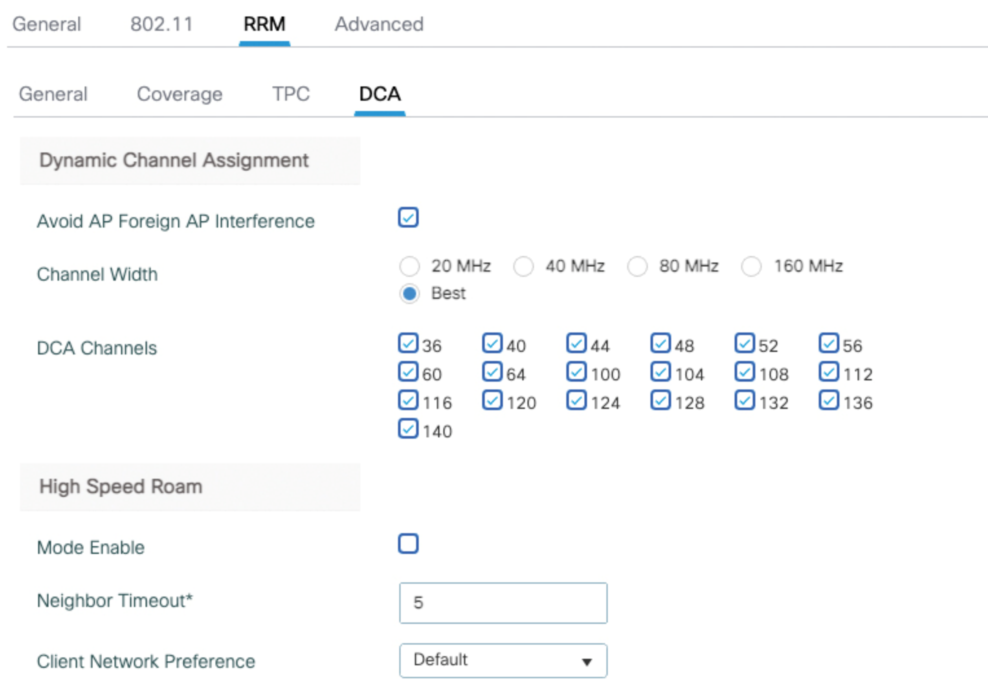


Figure 7-17 RF profile DCA configuration

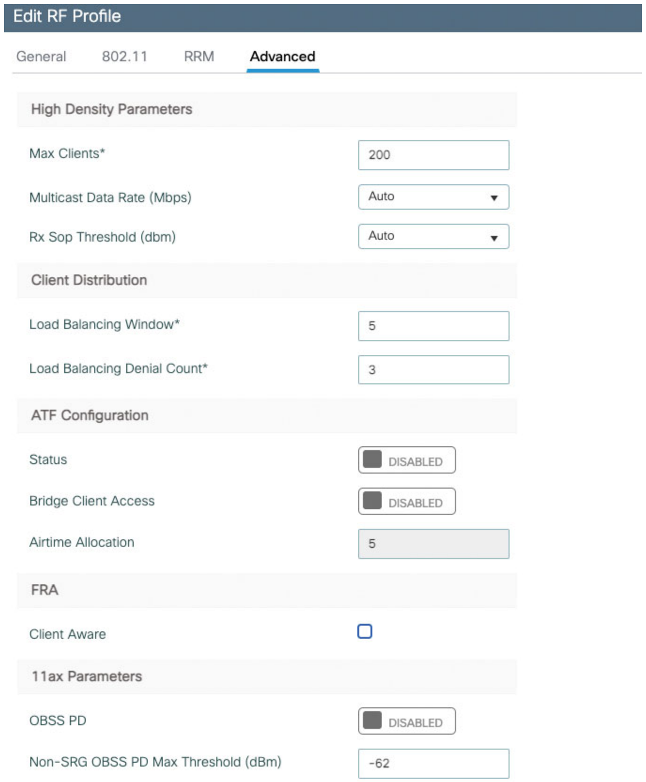


Figure 7-18 RF profile advanced configuration

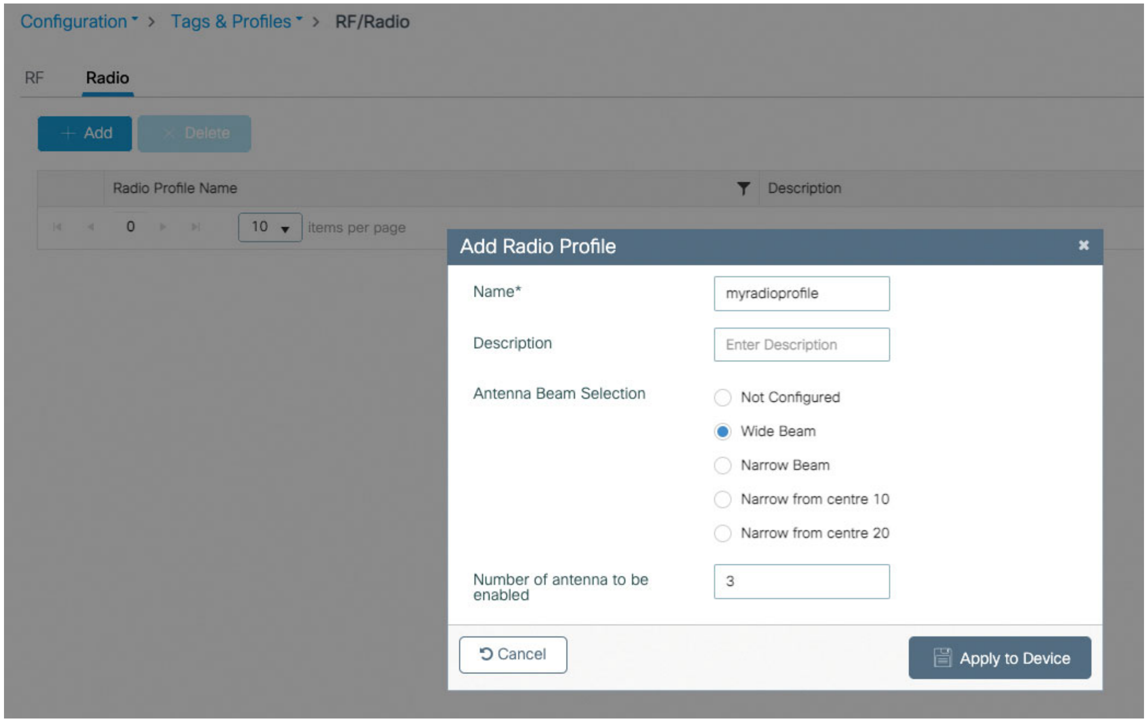


Figure 7-19 Radio profile configuration

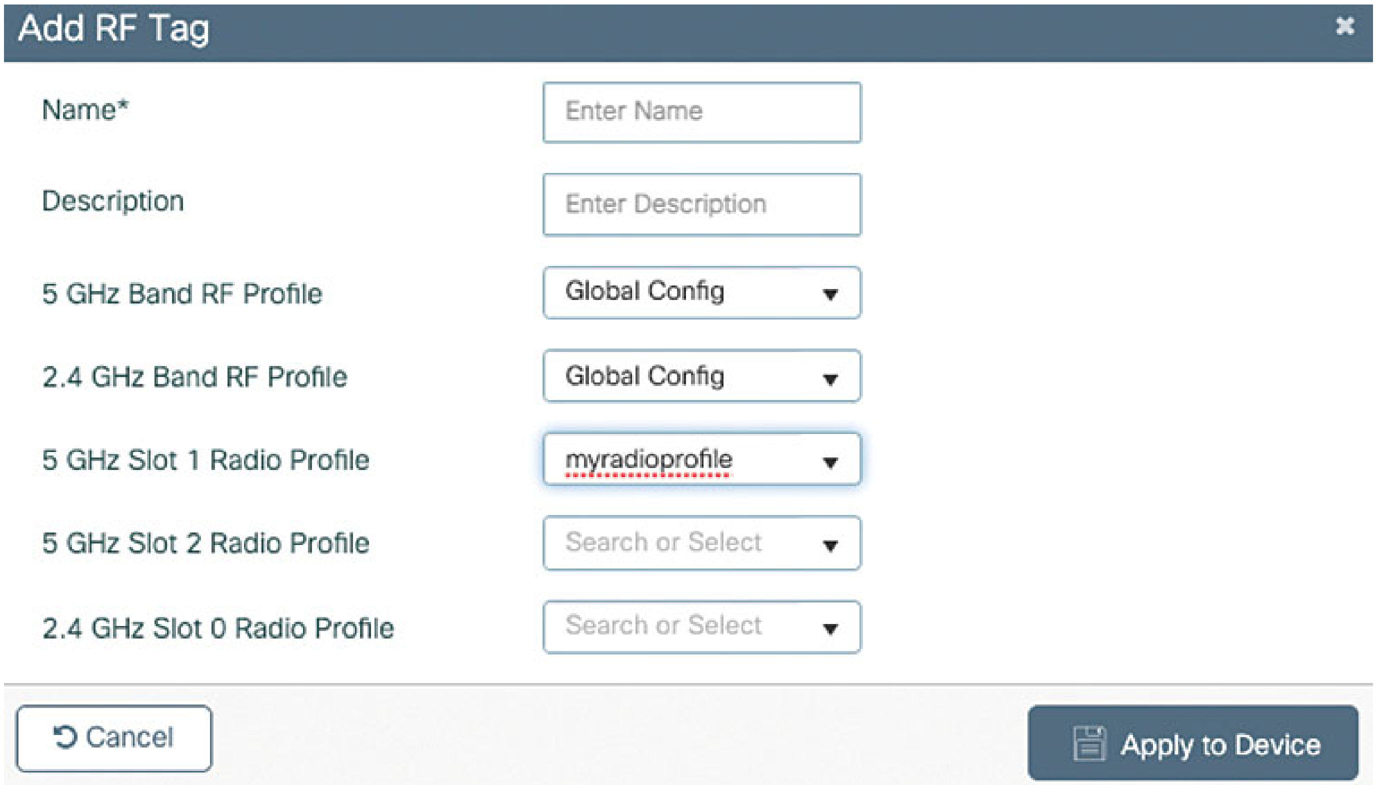


Figure 7-20 RF tag configuration

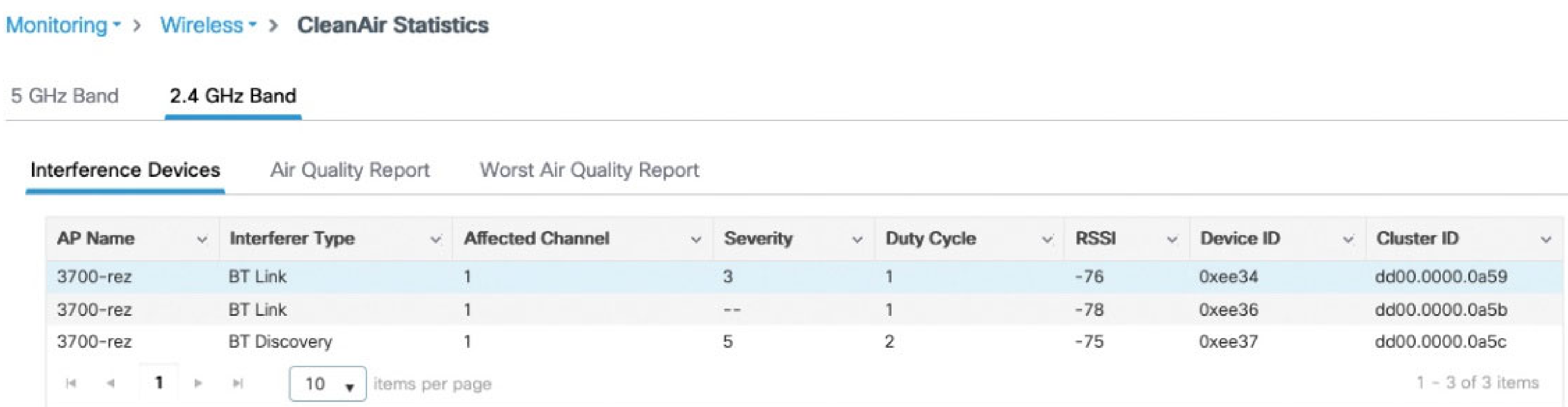


Figure 7-21 2.4 GHz CleanAir Interferer device report

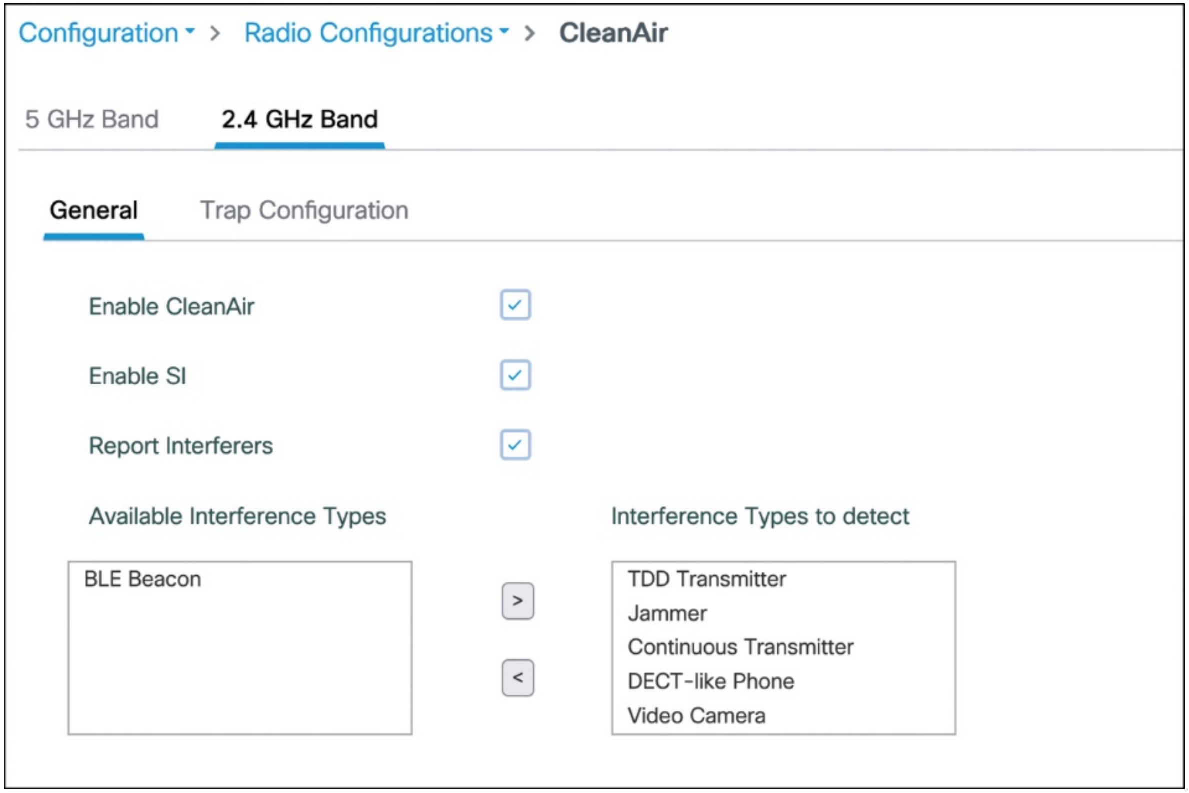


Figure 7-22 The CleanAir configuration page

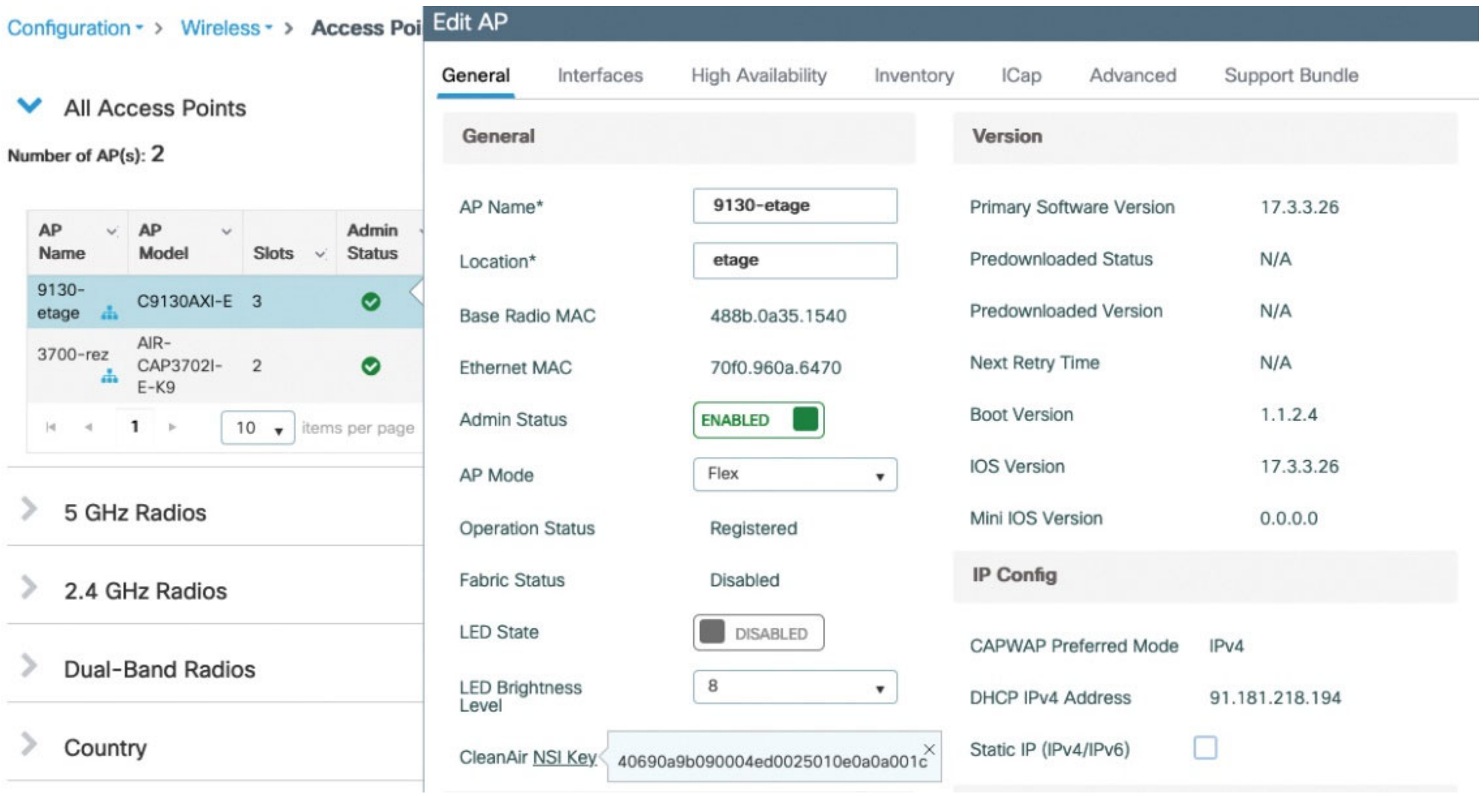


Figure 7-23 Obtaining the CleanAir key for a given AP

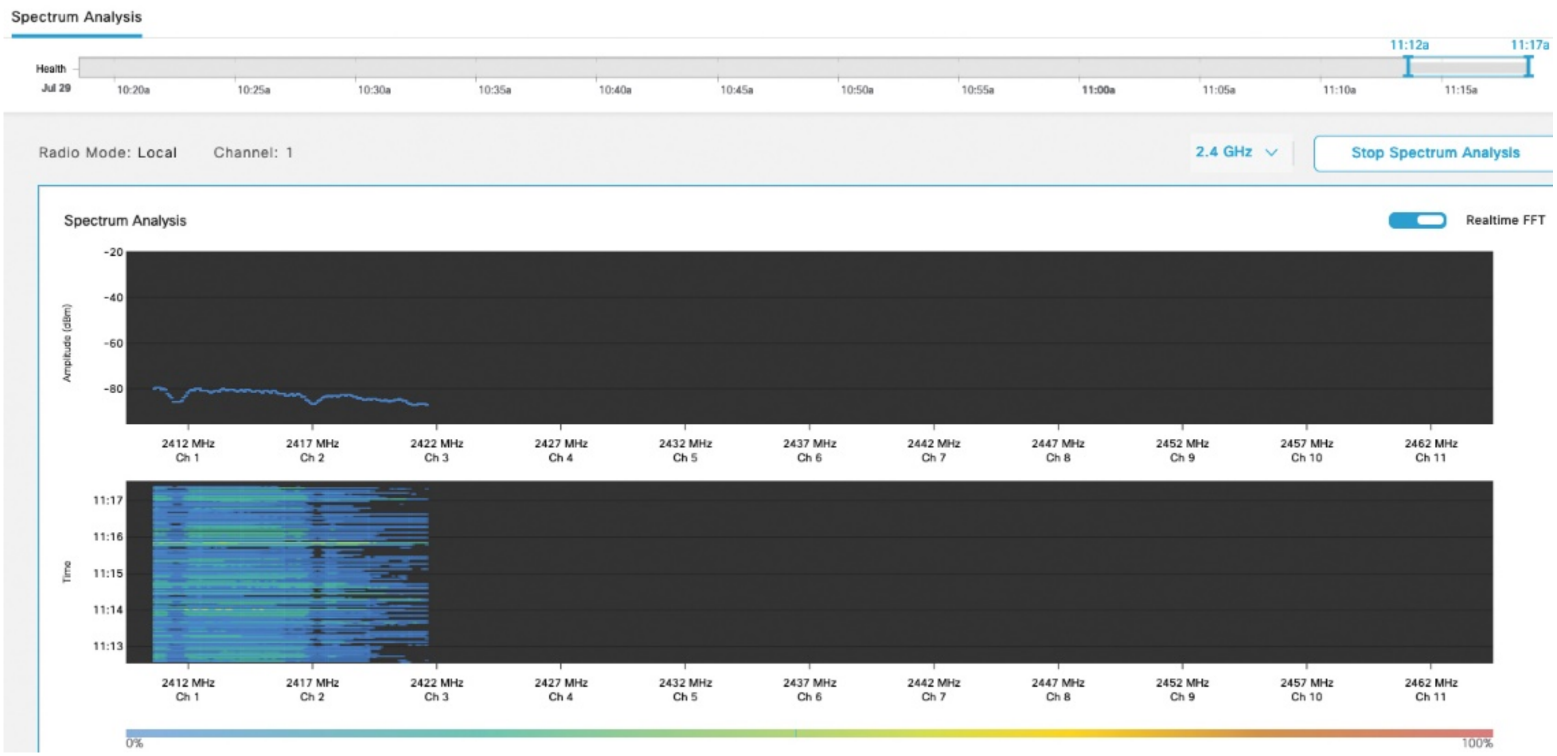


Figure 7-24 Cisco DNA Center spectrum live view

	Slot 0 (2.4 GHz)	Slot 1 (5 GHz)
Radio Type	802.11ax - 2.4 GHz	802.11ax - 5 GHz
Radio Role	Remote	Remote
Admin Status	Enabled	Enabled
Number of Clients	3	2
Current Channel	6	64
Power Level	*3/8 (9 dBm)	*1/7 (16 dBm)
Channel Utilization	15%	1%
Transmit Utilization	0%	1%
Receive Utilization	0%	0%

Figure 7-25 Radio statistics of an AP from the C9800 web interface

From *Understanding and Troubleshooting Cisco Catalyst 9800 Series Wireless Controller by* Simone Arena, Francisco Sedano Crippa, Nicolas Darchis and Sudha Katgeri(9780137492329) Cisco Press

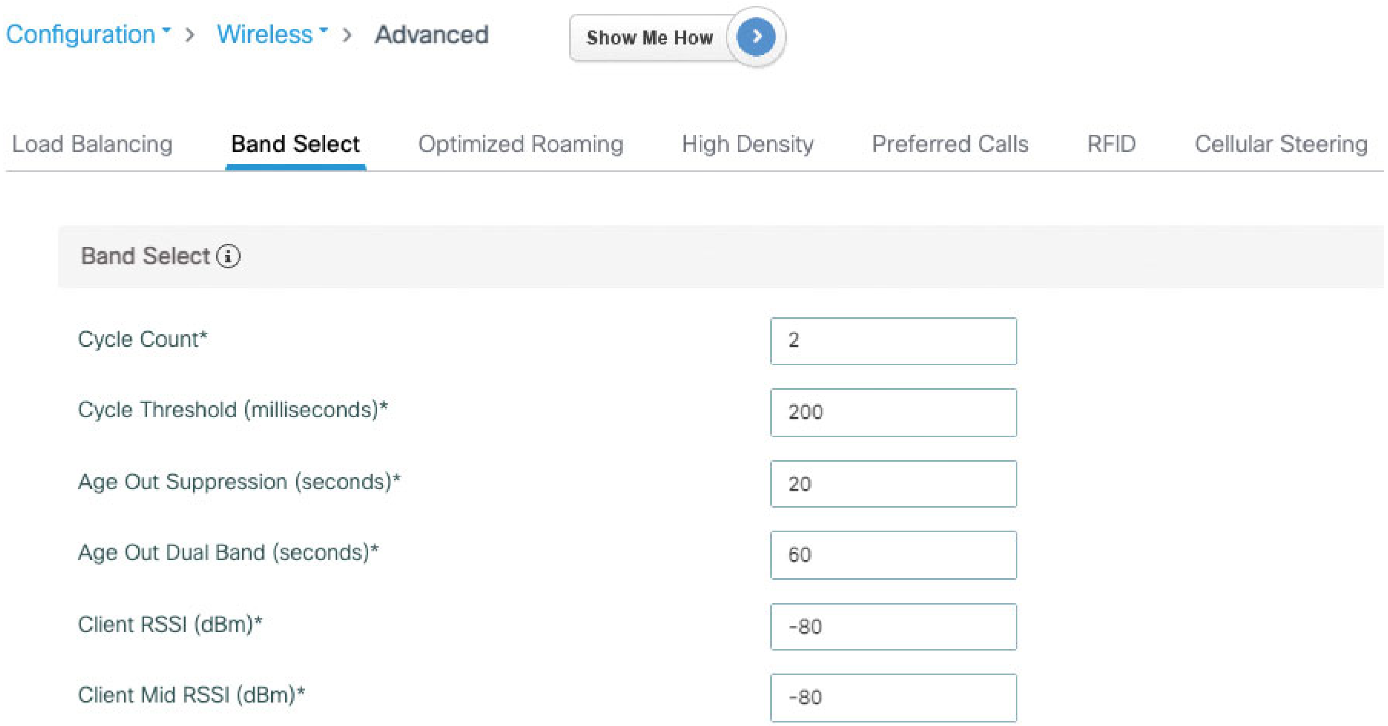


Figure 7-26 Band Select global configuration

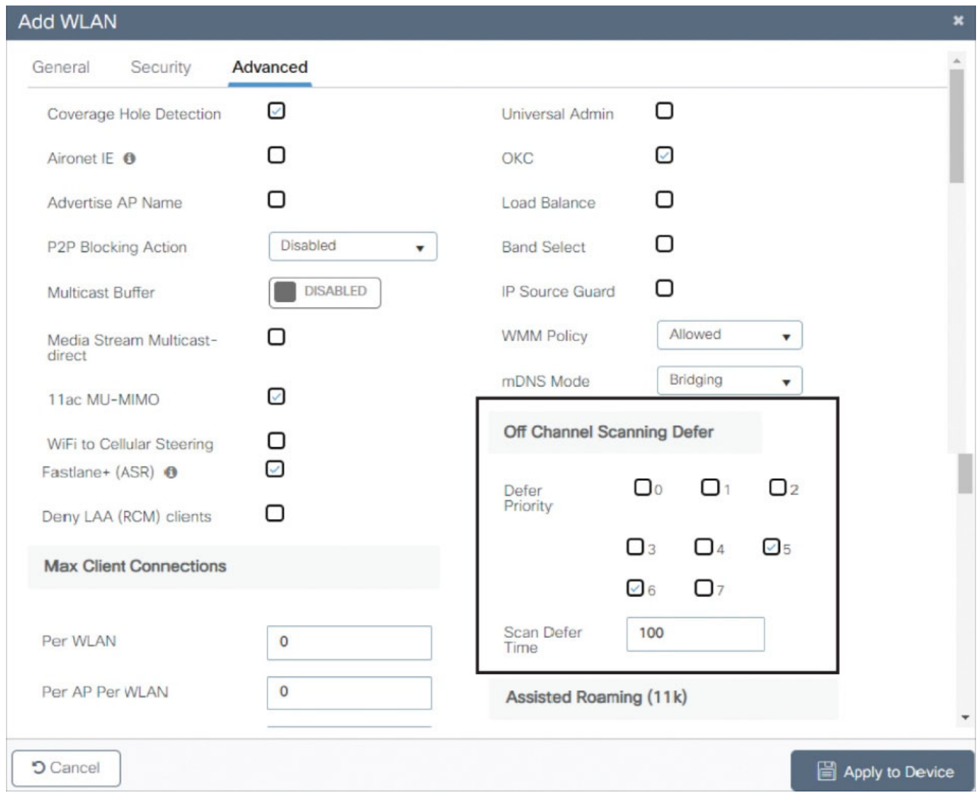


Figure 7-27 Off-Channel Scanning Defer setting

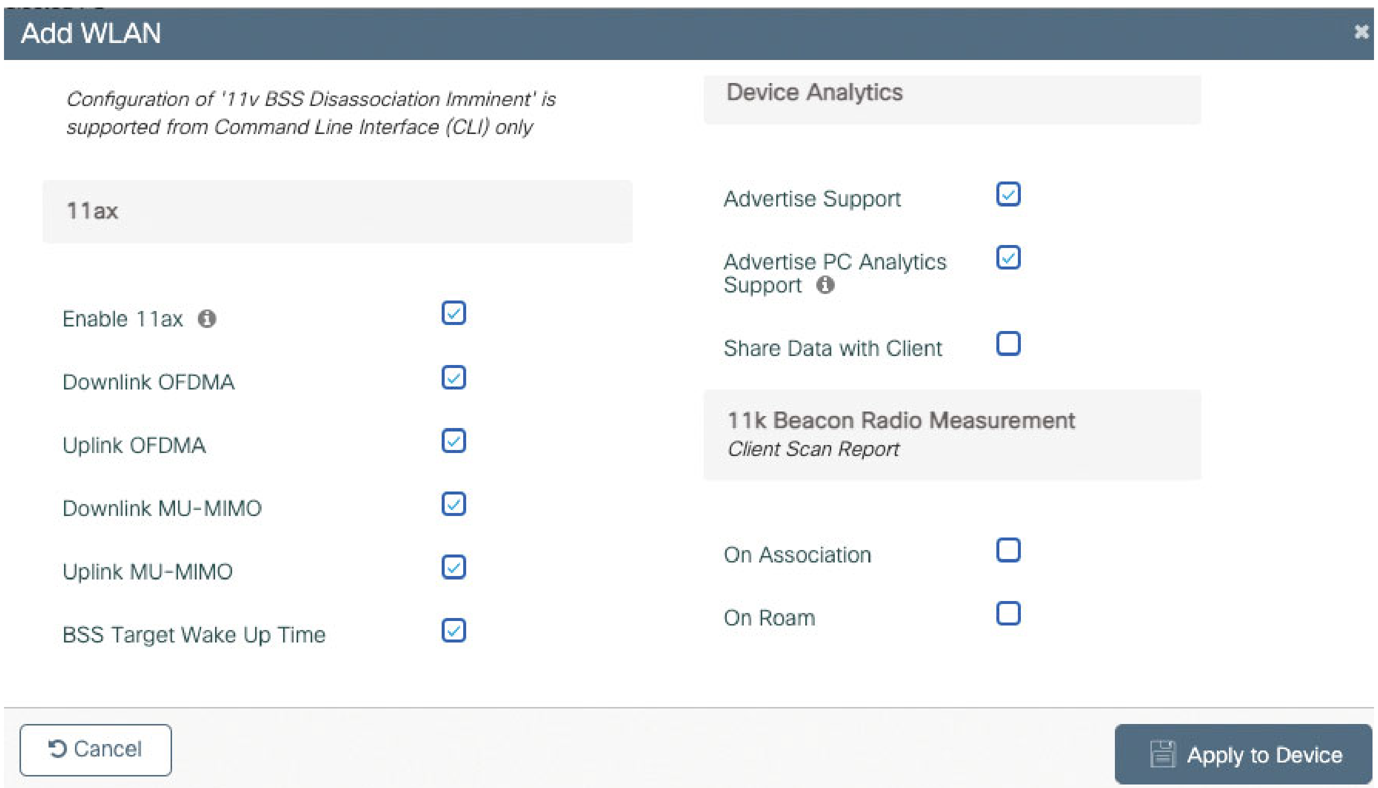


Figure 7-28 WLAN 11ax features configuration

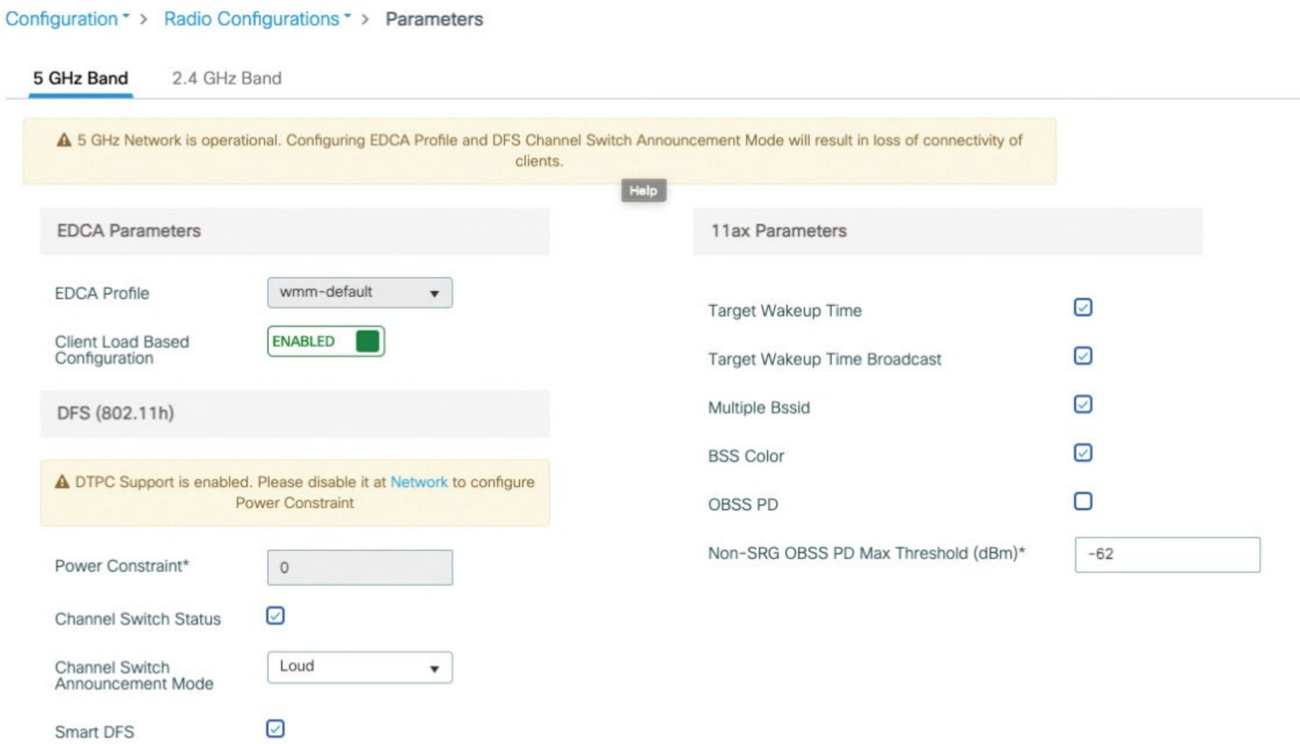


Figure 7-29 Global 11ax features configuration

Figure 7-30 OBSS coloring configuration

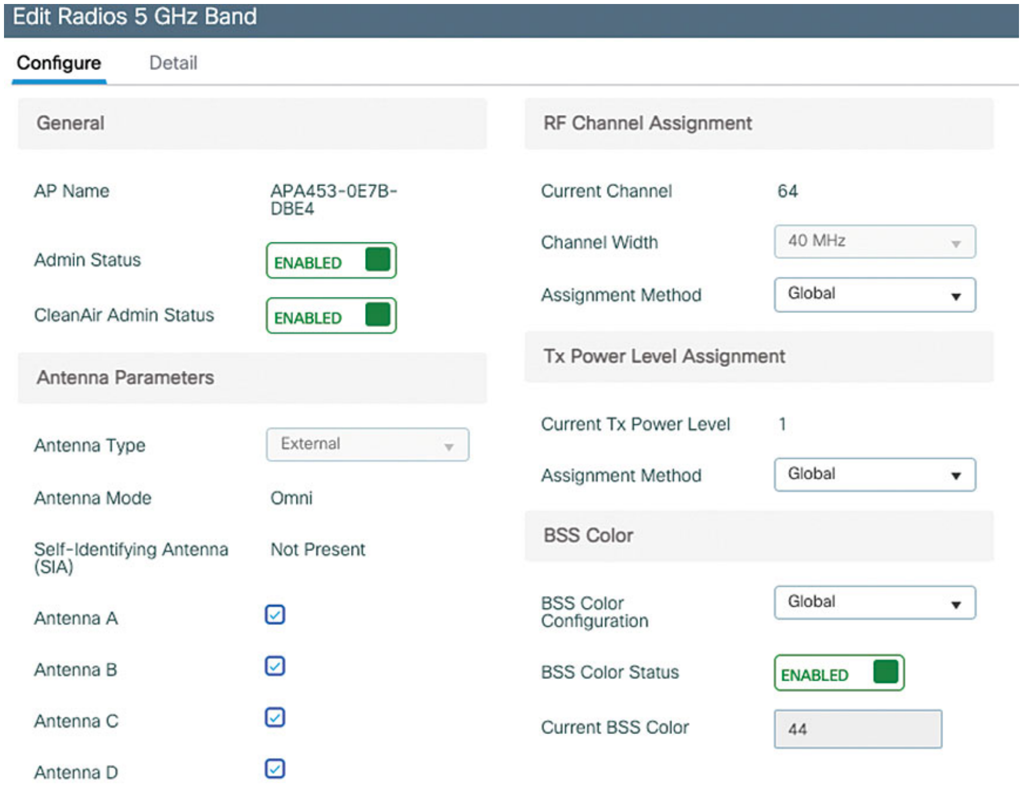


Figure 7-31 AP-specific BSS color options

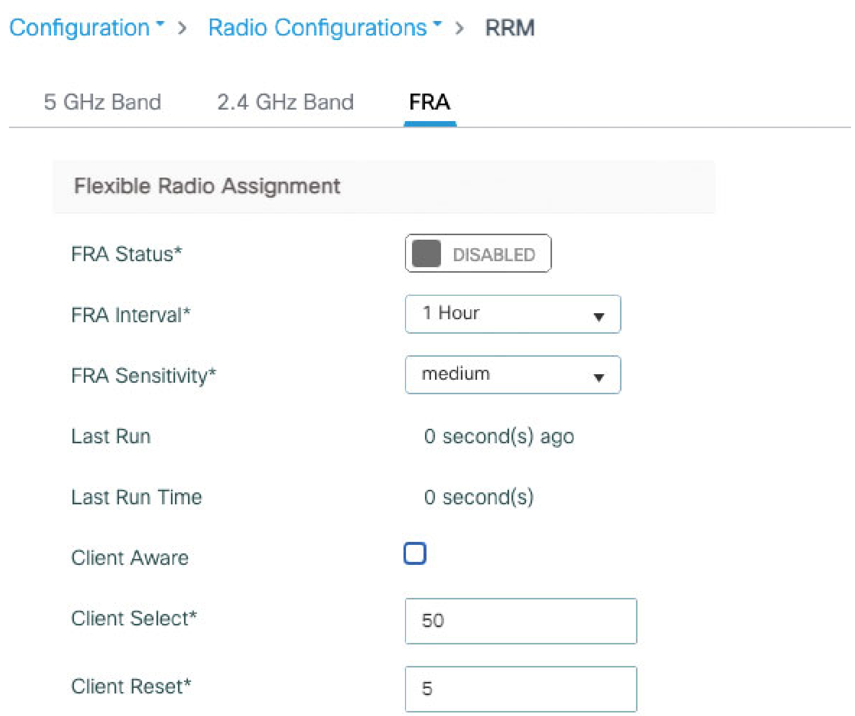


Figure 7-32 FRA settings on the Catalyst 9800 controller

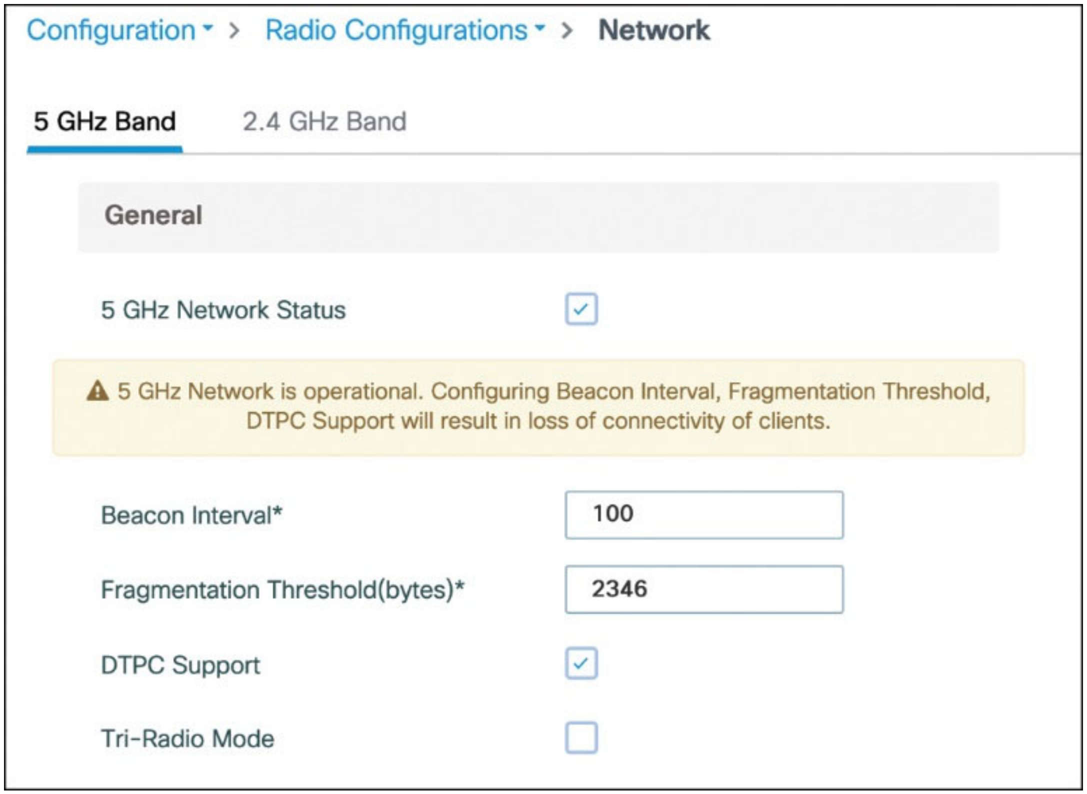


Figure 7-33 Global tri-radio setting

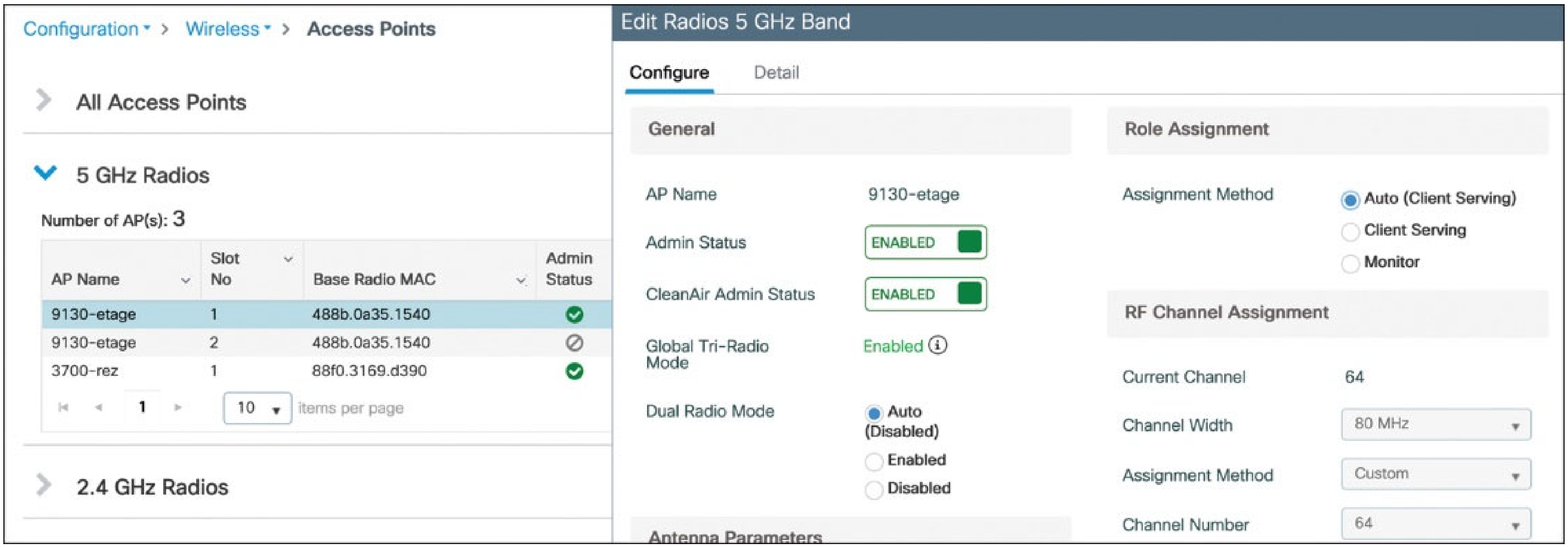


Figure 7-34 Global tri-radio setting

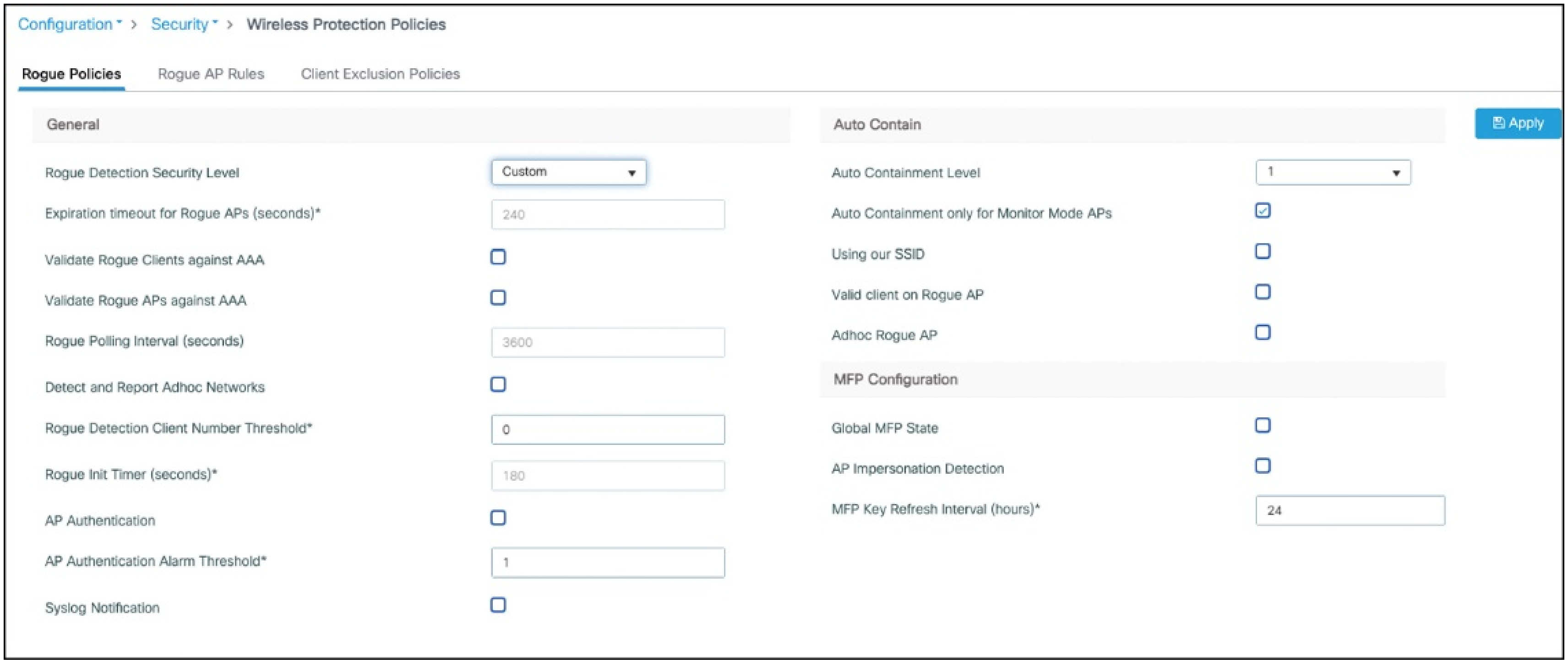


Figure 7-35 Rogue detection settings

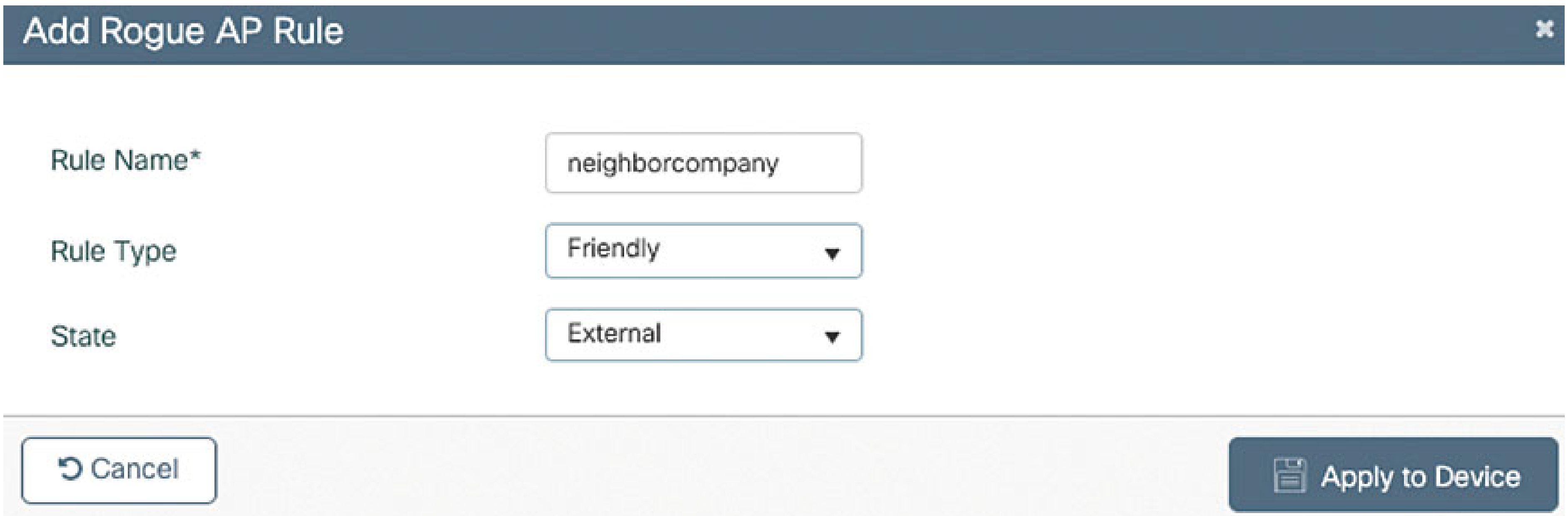


Figure 7-36 A friendly rogue AP rule

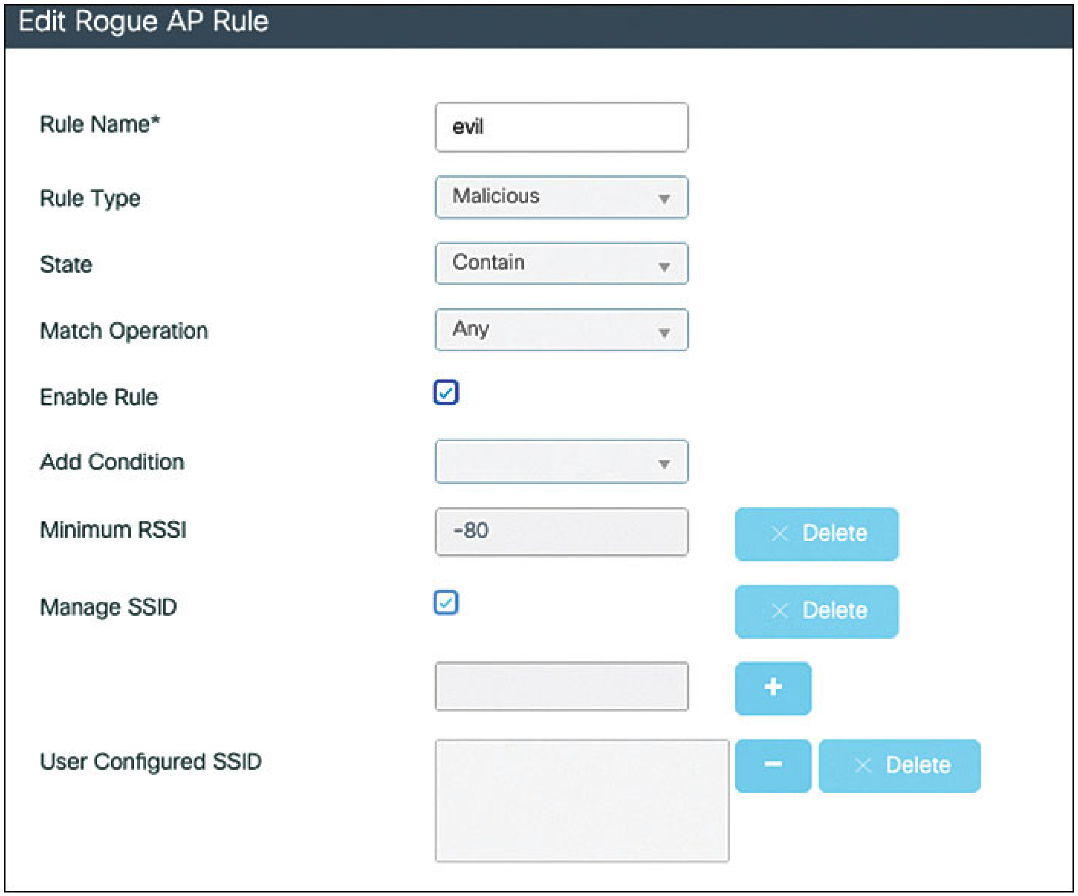


Figure 7-37 A malicious rogue AP rule

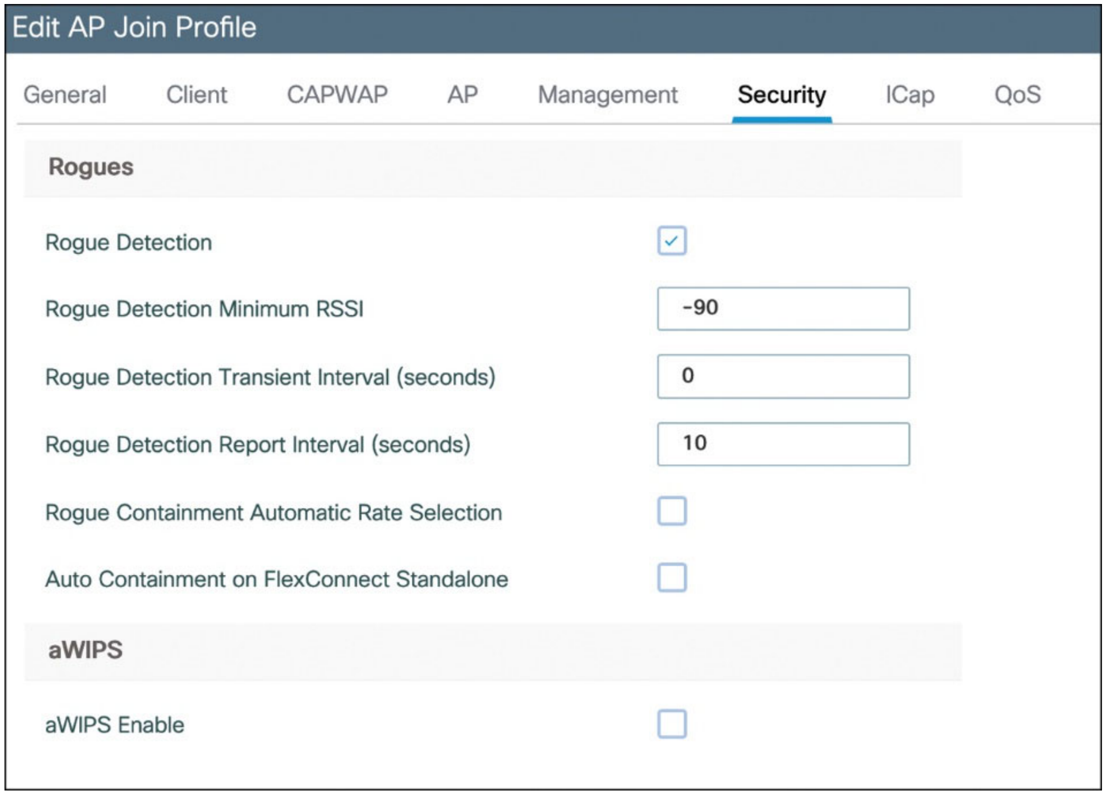


Figure 7-38 AP join rogue detection settings

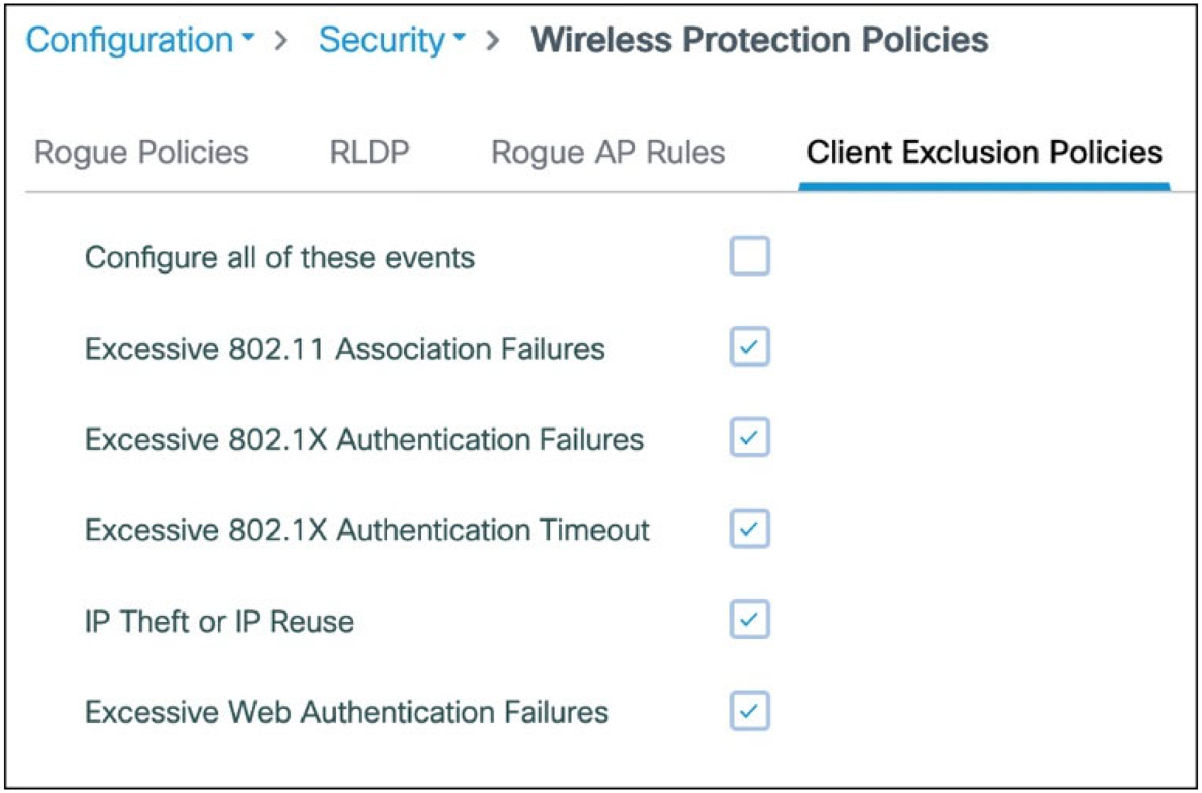


Figure 7-39 Client exclusion policies on the C9800

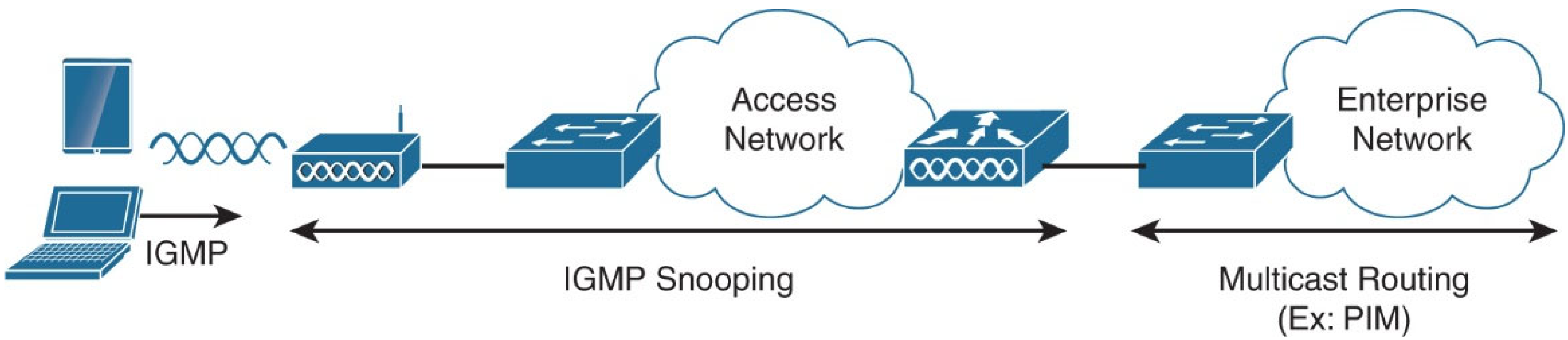


Figure 8-1 Multicast in wireless

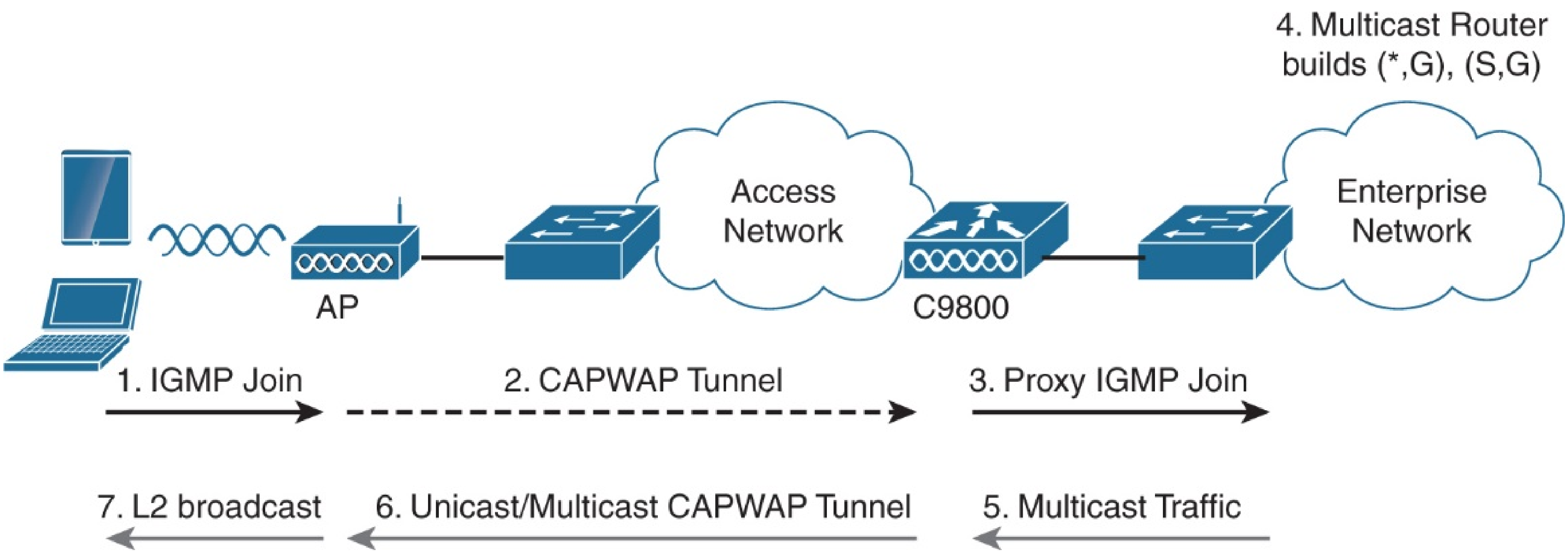


Figure 8-2 Multicast packet flow in wireless

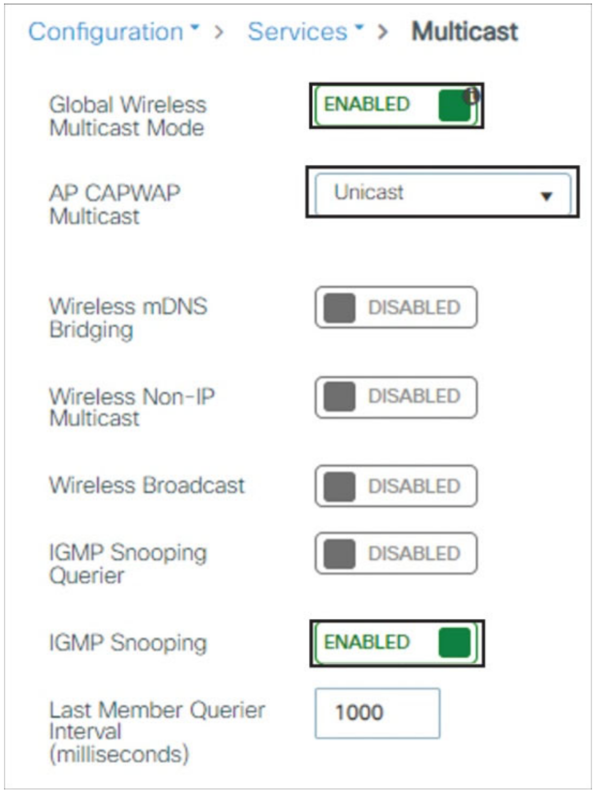


Figure 8-3 Enabling Multicast and MoU/MoM on the C9800

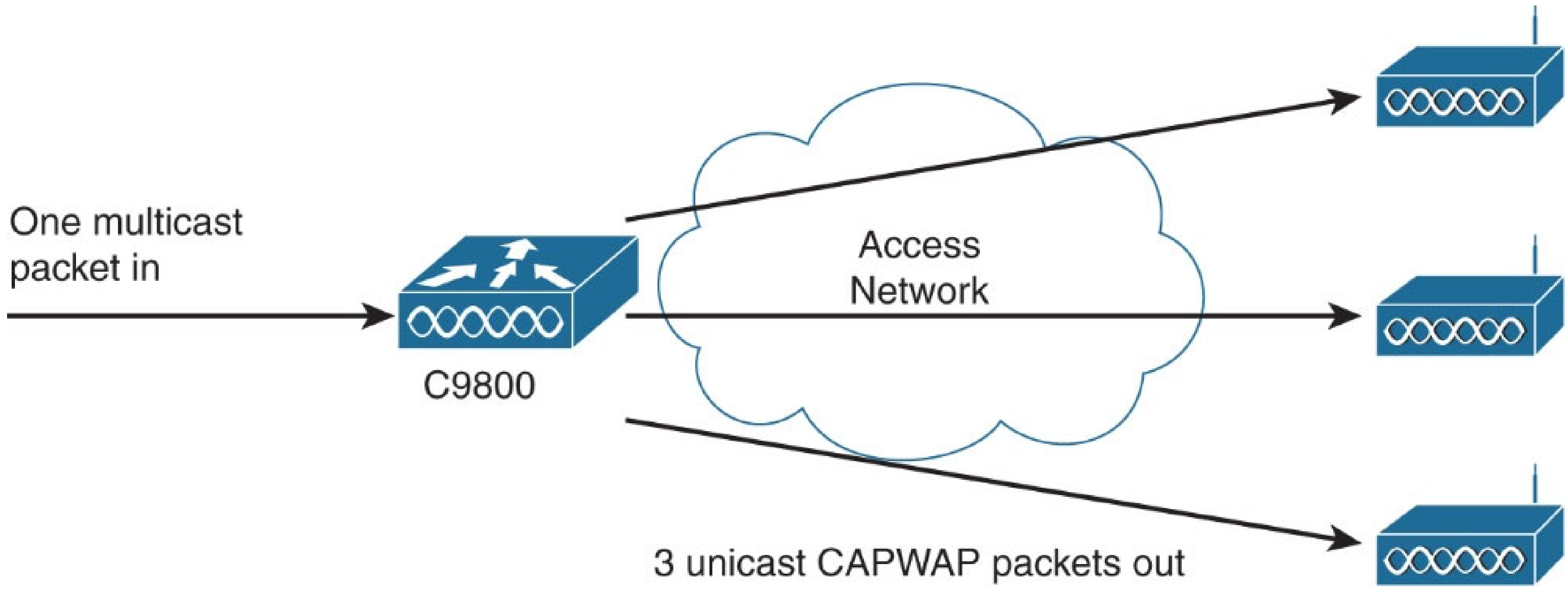


Figure 8-4 MoU: the C9800 creating copies for each multicast packet

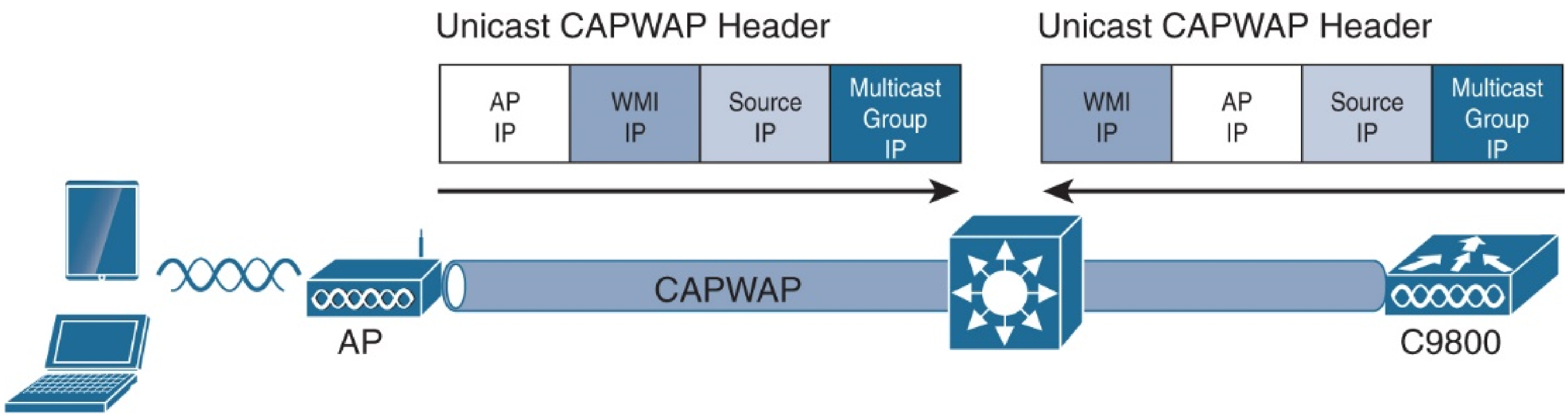


Figure 8-5 Packet format in MoU mode

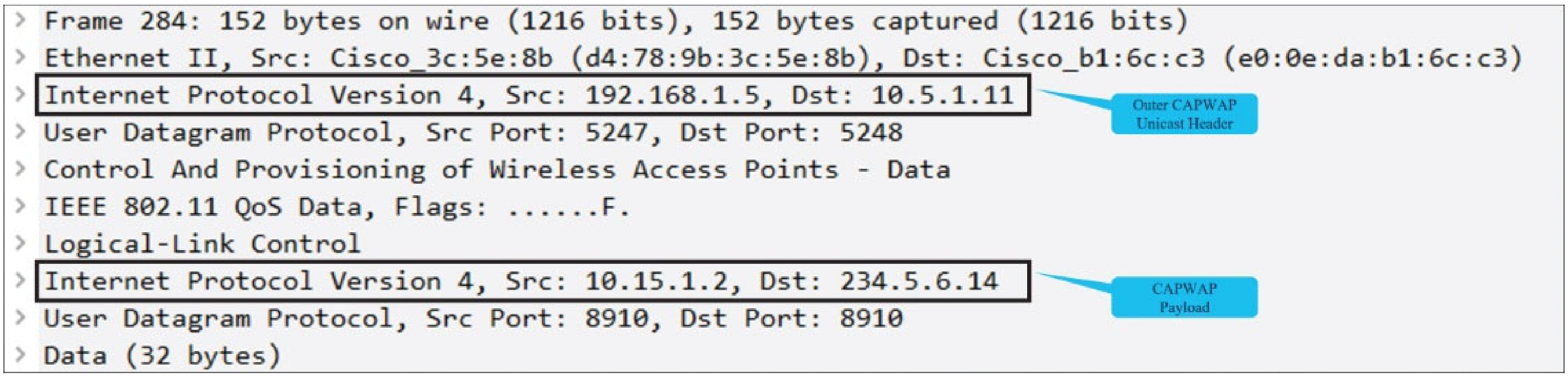


Figure 8-6 Multicast packet snippet from the C9800 to the AP in MoM mode

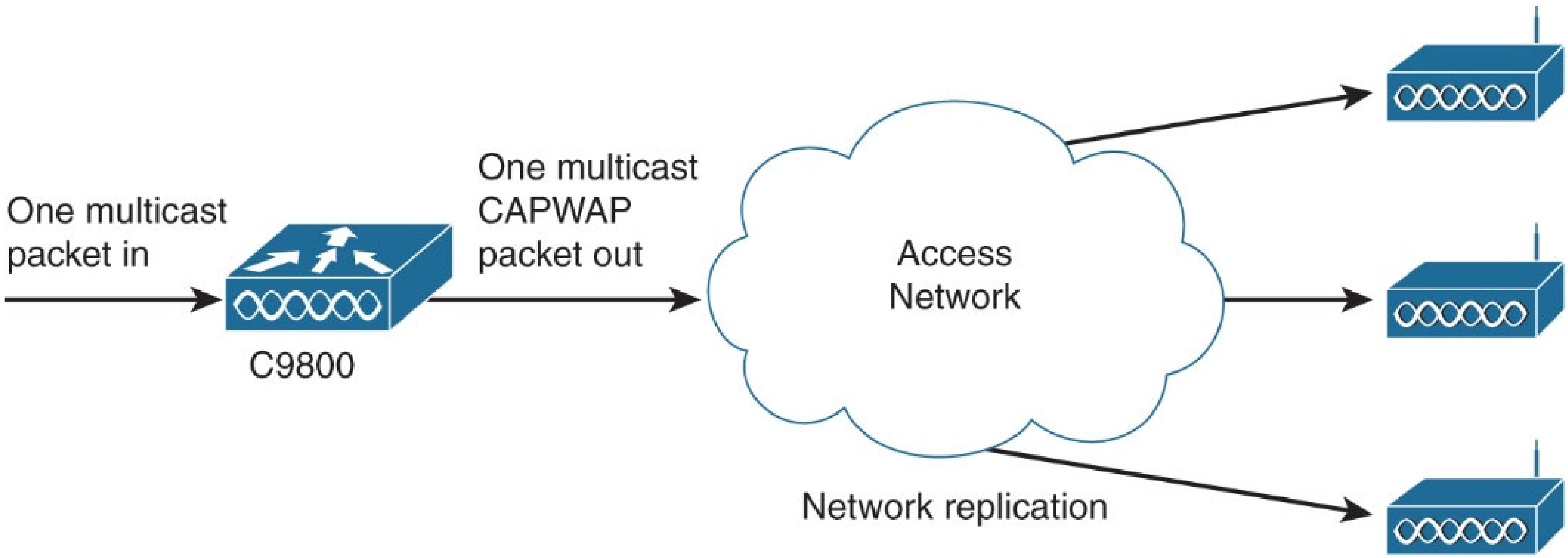


Figure 8-7 MoM: Network creates copies of multicast packets

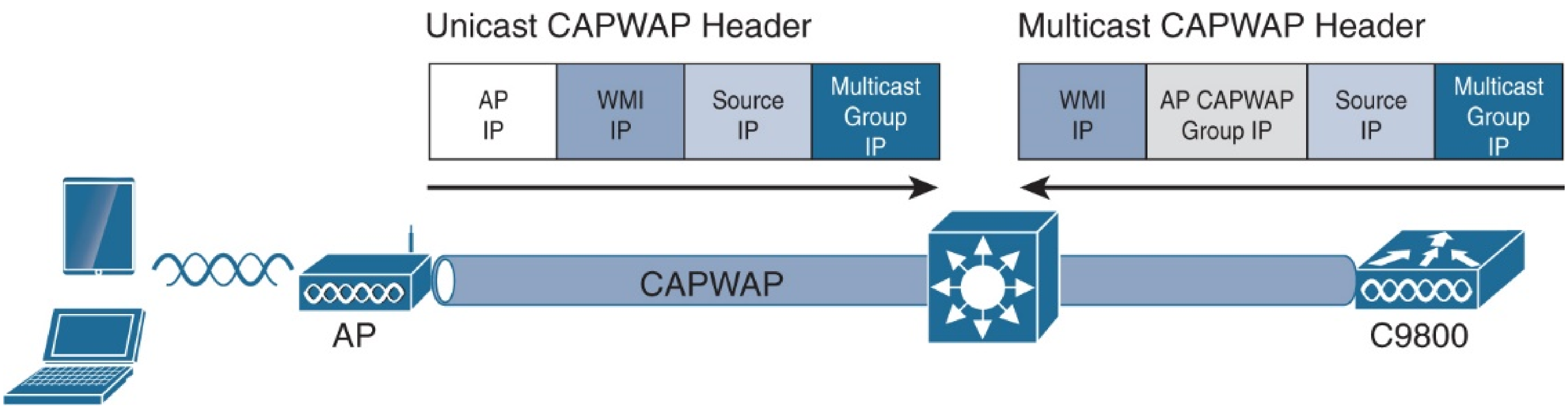


Figure 8-8 Packet format in MoM

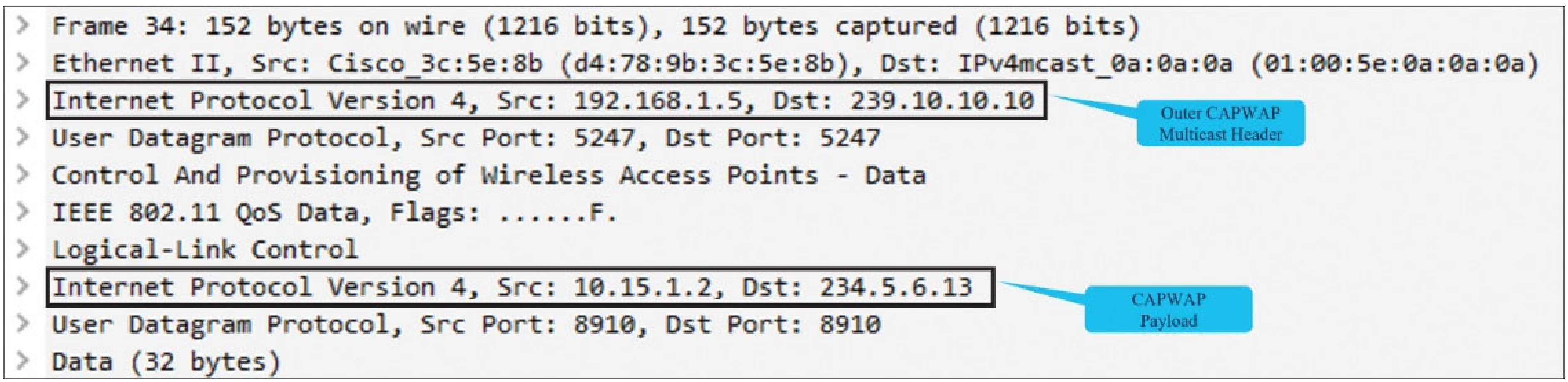


Figure 8-9 Multicast packet snippet from the C9800 to the AP in MoU mode

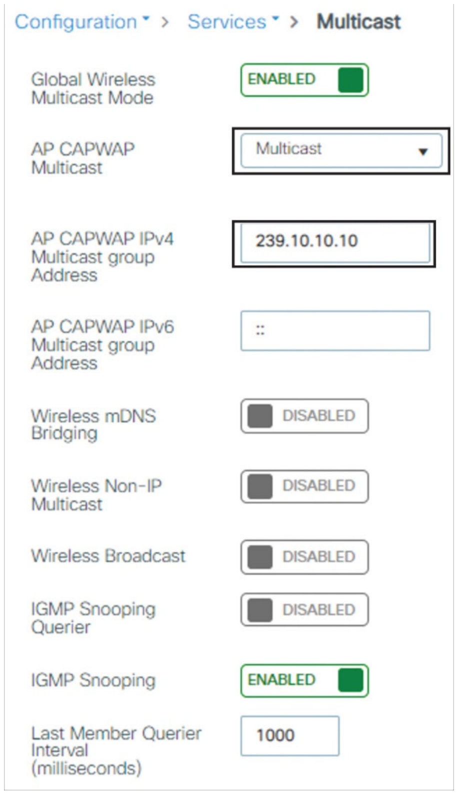


Figure 8-10 Configuring MoM on the C9800

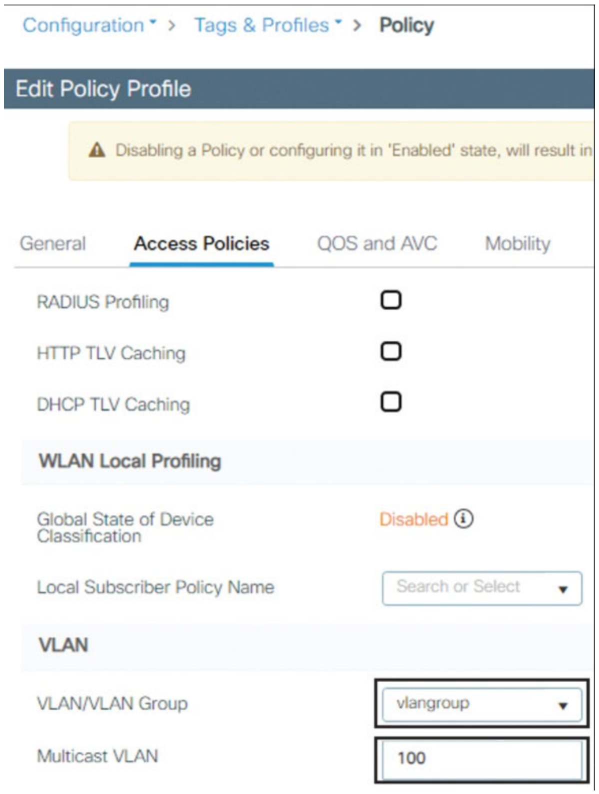
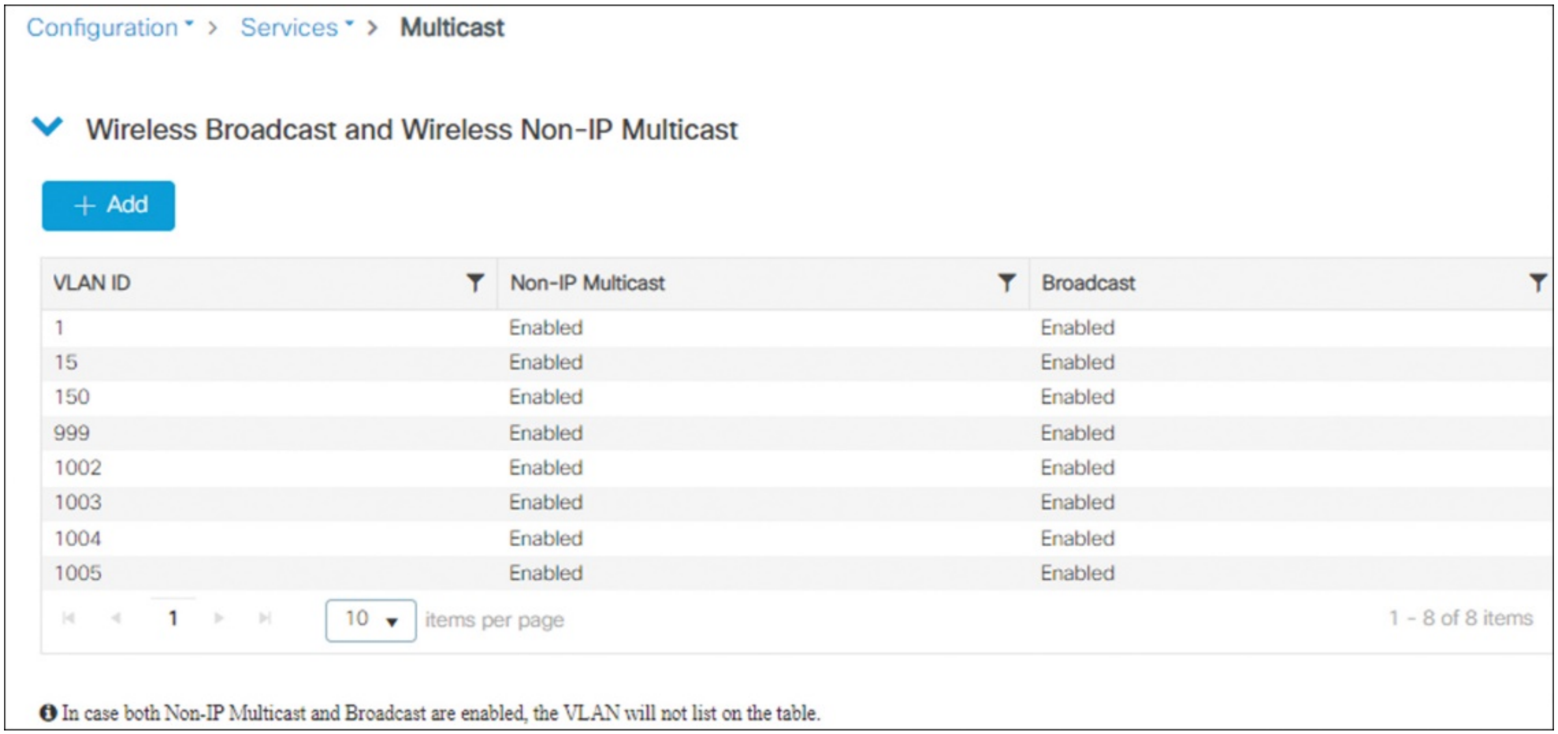


Figure 8-11 Configuring VLAN Select or multicast VLAN on the C9800 wireless policy profile



VLAN ID	Non-IP Multicast	Broadcast
1	Enabled	Enabled
15	Enabled	Enabled
150	Enabled	Enabled
999	Enabled	Enabled
1002	Enabled	Enabled
1003	Enabled	Enabled
1004	Enabled	Enabled
1005	Enabled	Enabled

Figure 8-12 Default settings of wireless broadcast and non-IP multicast per VLAN when wireless multicast is enabled

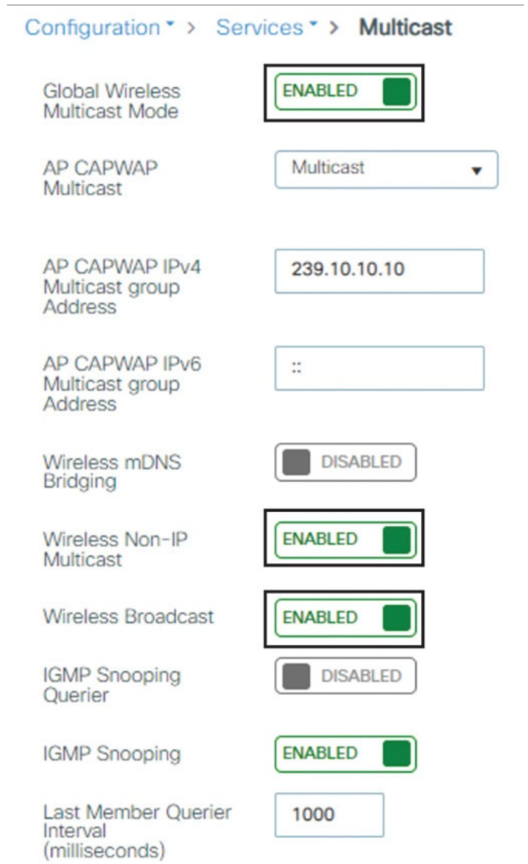


Figure 8-13 Configuring wireless broadcast and non-IP multicast forwarding

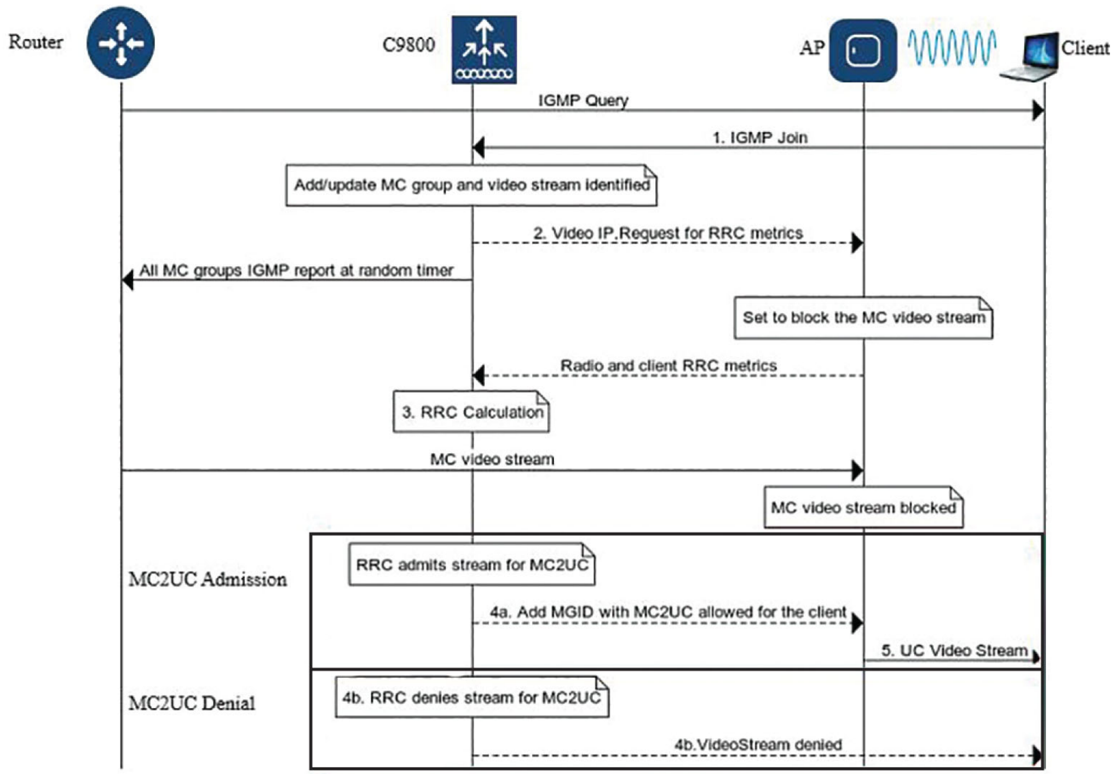


Figure 8-14 Media Stream packet flow

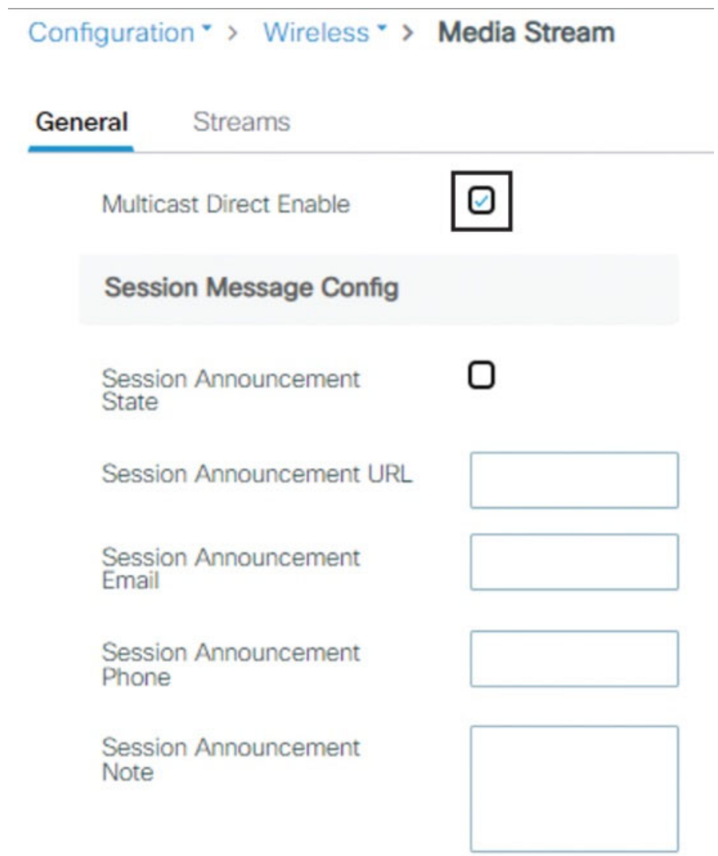


Figure 8-15 Configuring Media Stream globally

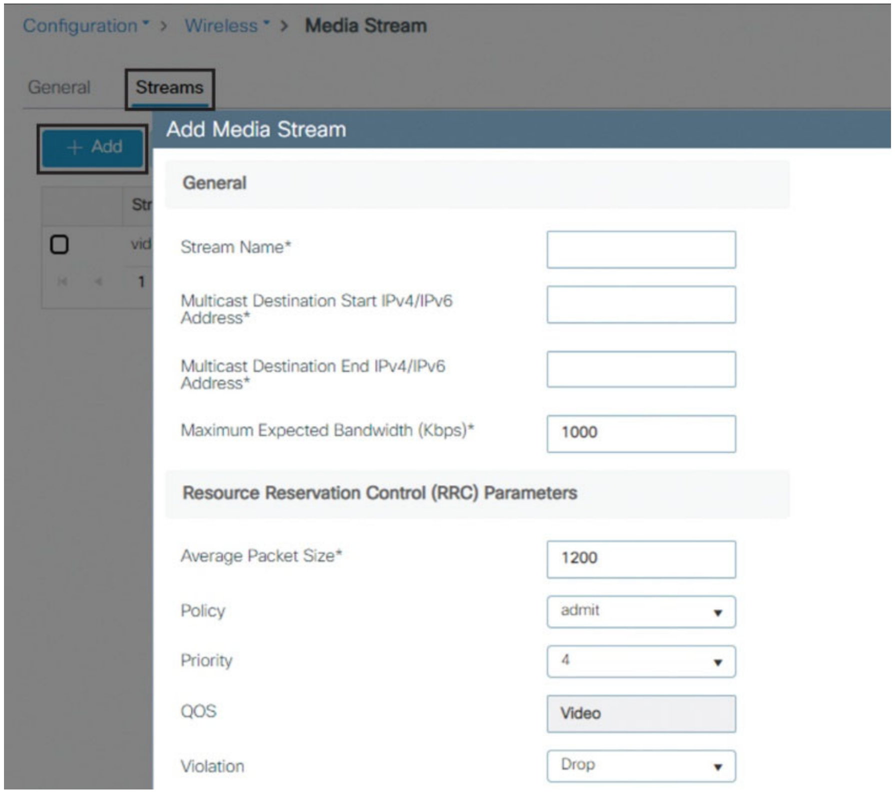


Figure 8-16 Configuring multicast stream details with their characteristics

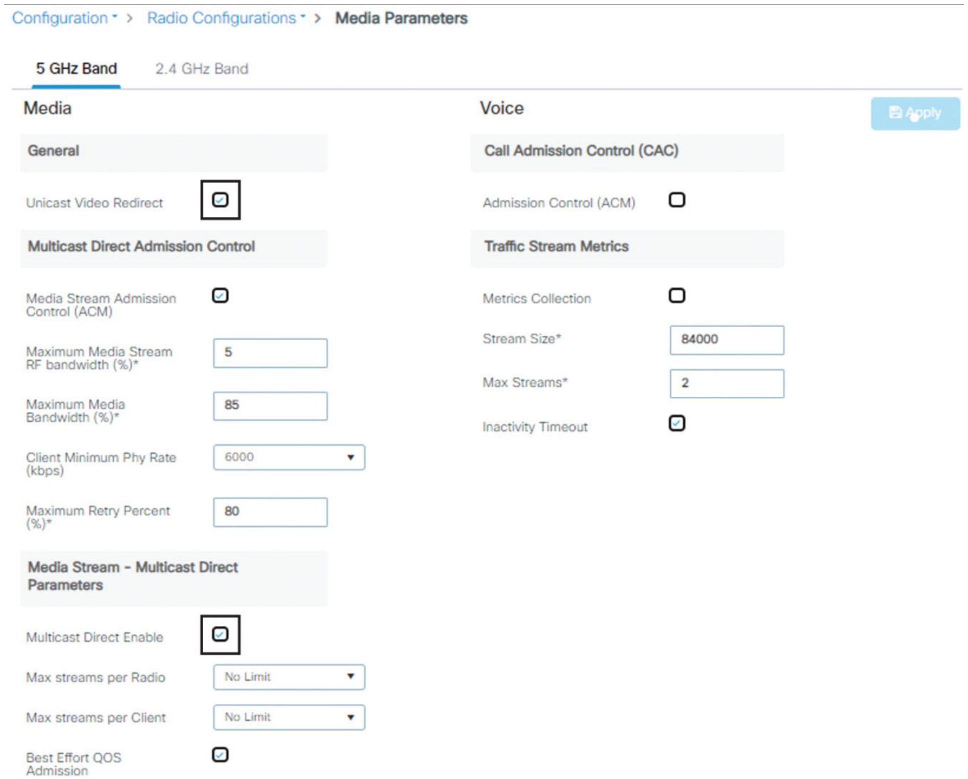


Figure 8-17 Configuring Media Stream on 2.4 GHz and 5 GHz bands

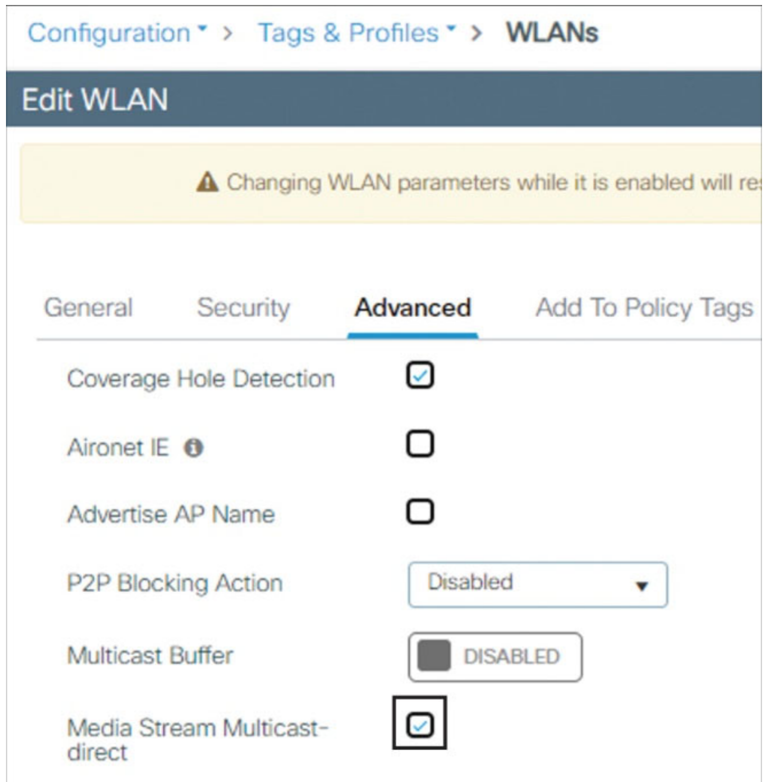


Figure 8-18 Enabling Media Stream on a client WLAN profile

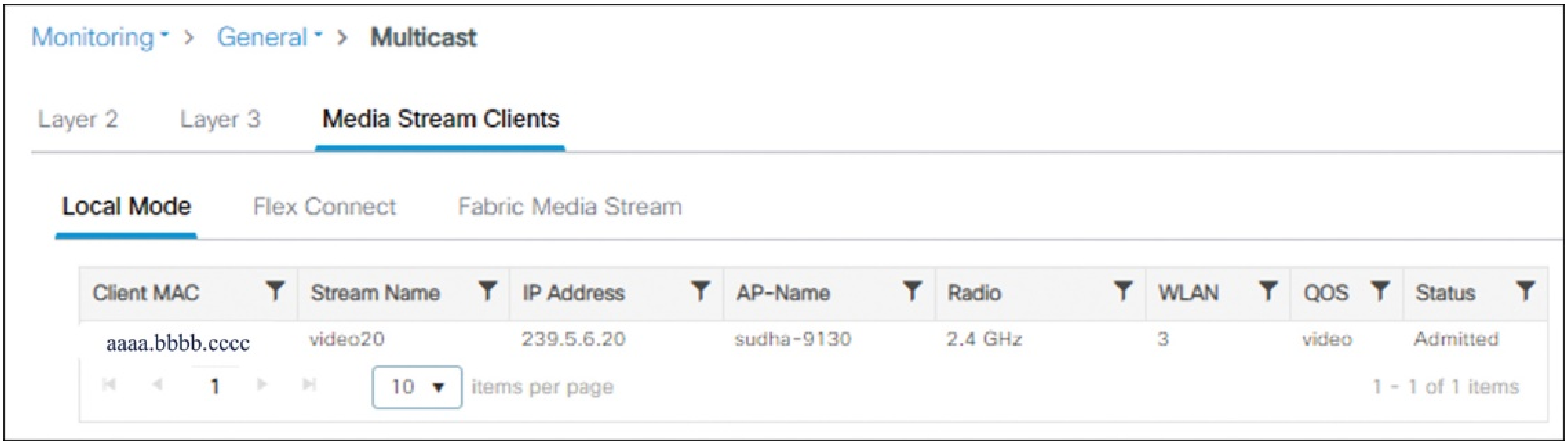


Figure 8-19 Monitoring a client admitted to the MC2UC Media Stream

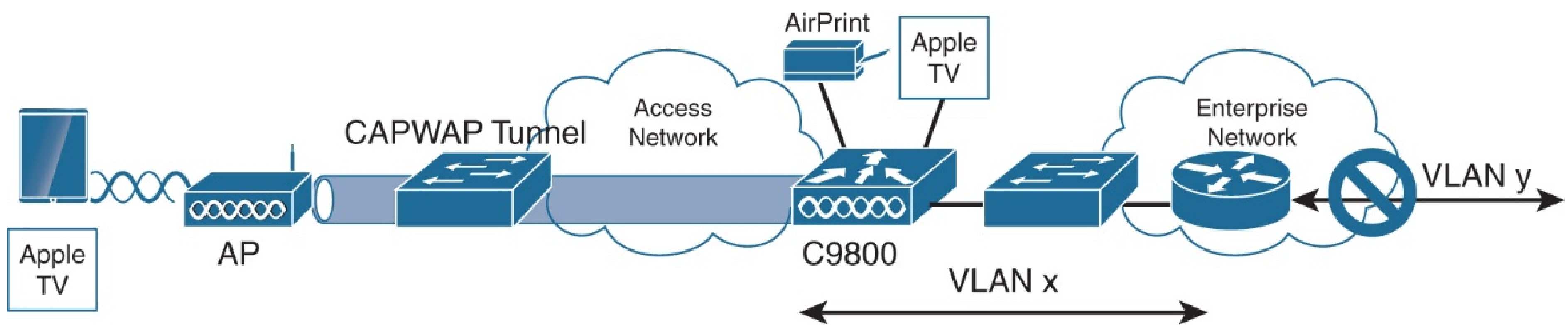


Figure 8-20 Default mDNS bridging in action

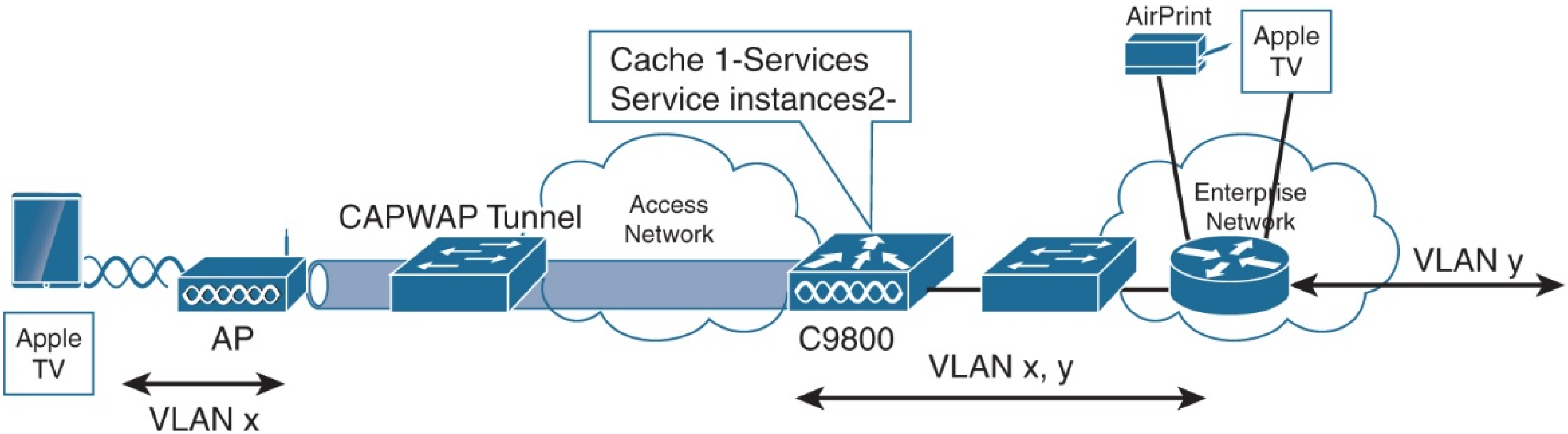


Figure 8-21 mDNS gateway in action

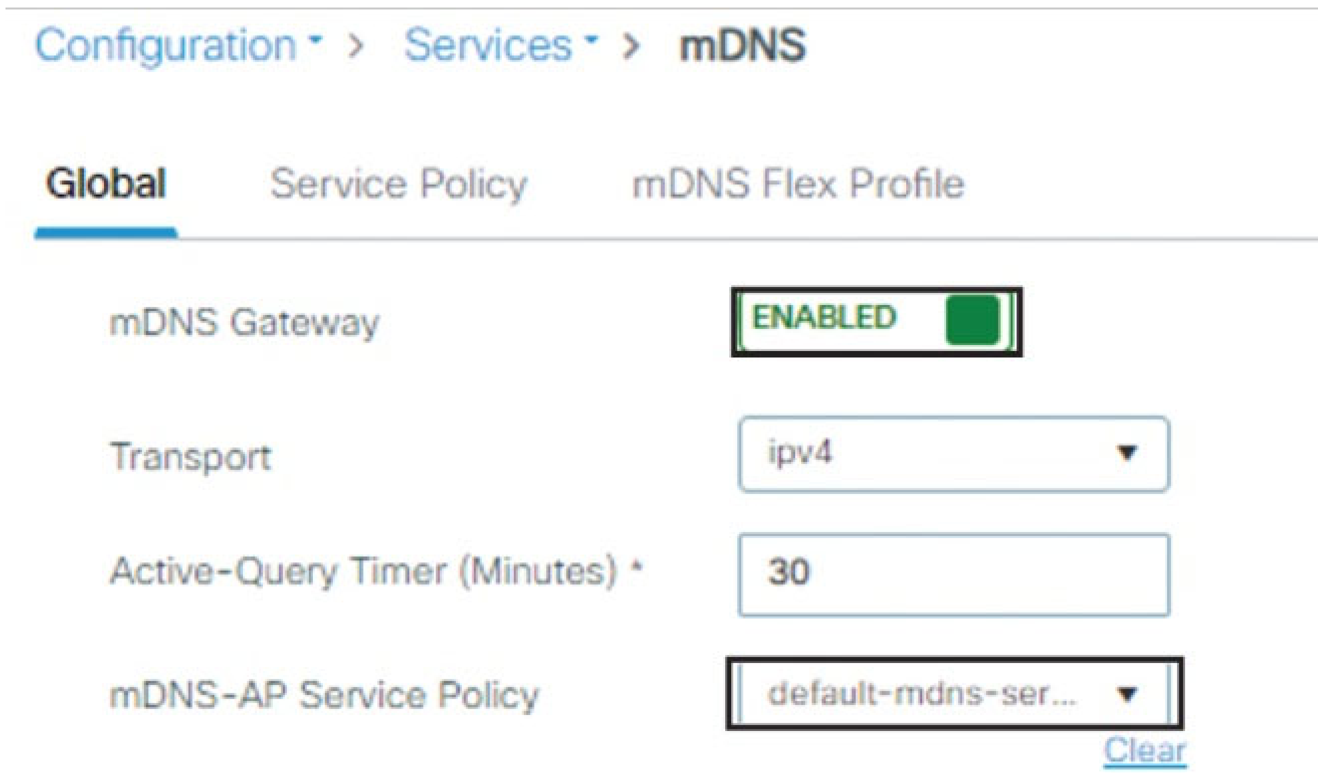


Figure 8-22 Configuring the C9800 as mDNS gateway globally

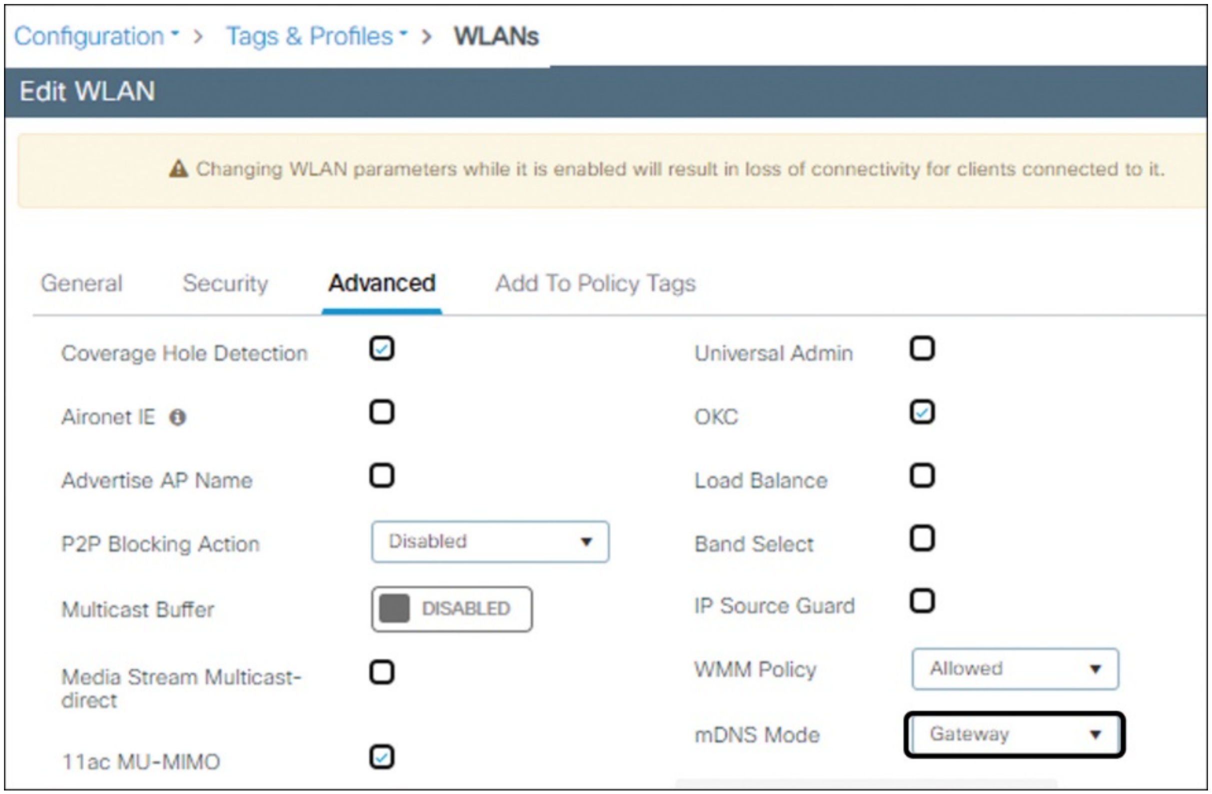


Figure 8-23 Enabling the MDNS gateway on the WLAN

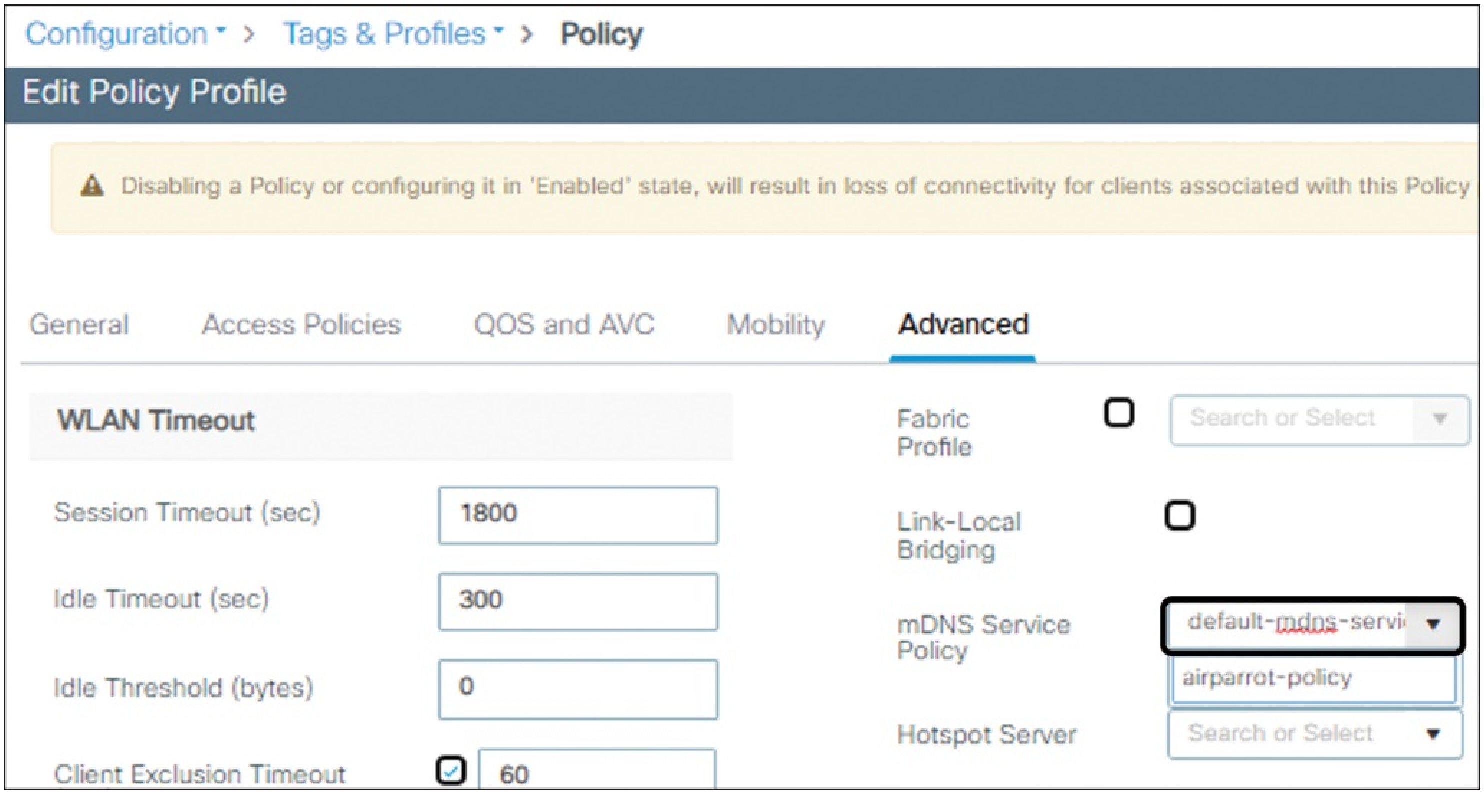


Figure 8-24 Configuring an mDNS service policy on a wireless policy profile

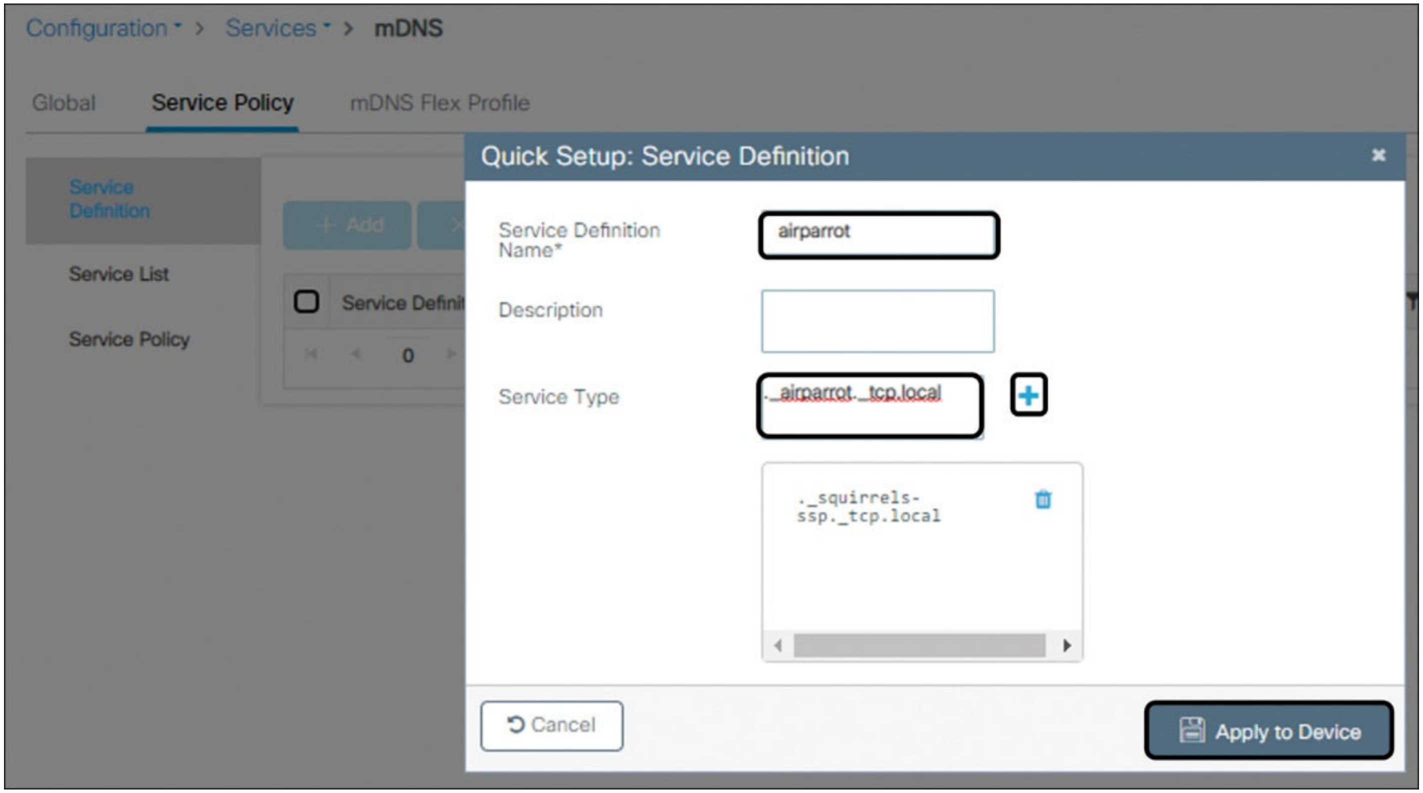


Figure 8-25 Creating a custom service definition

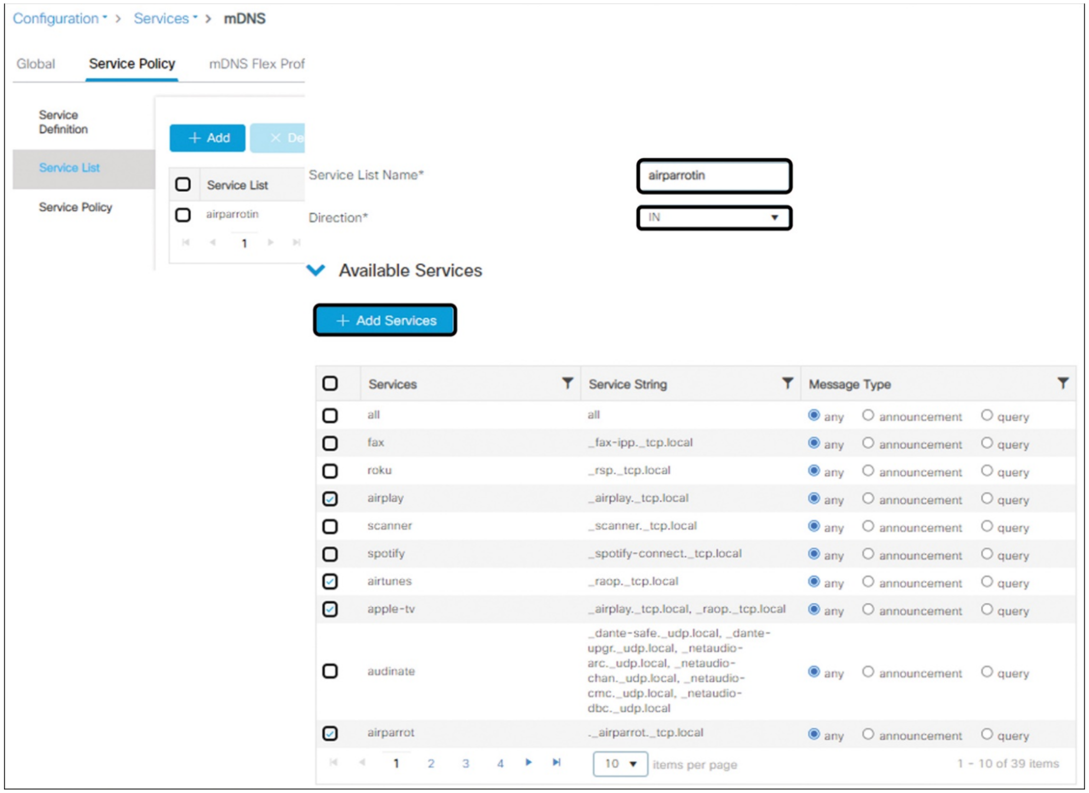


Figure 8-26 Creating a service list mapping custom service definitions and services from the master service list

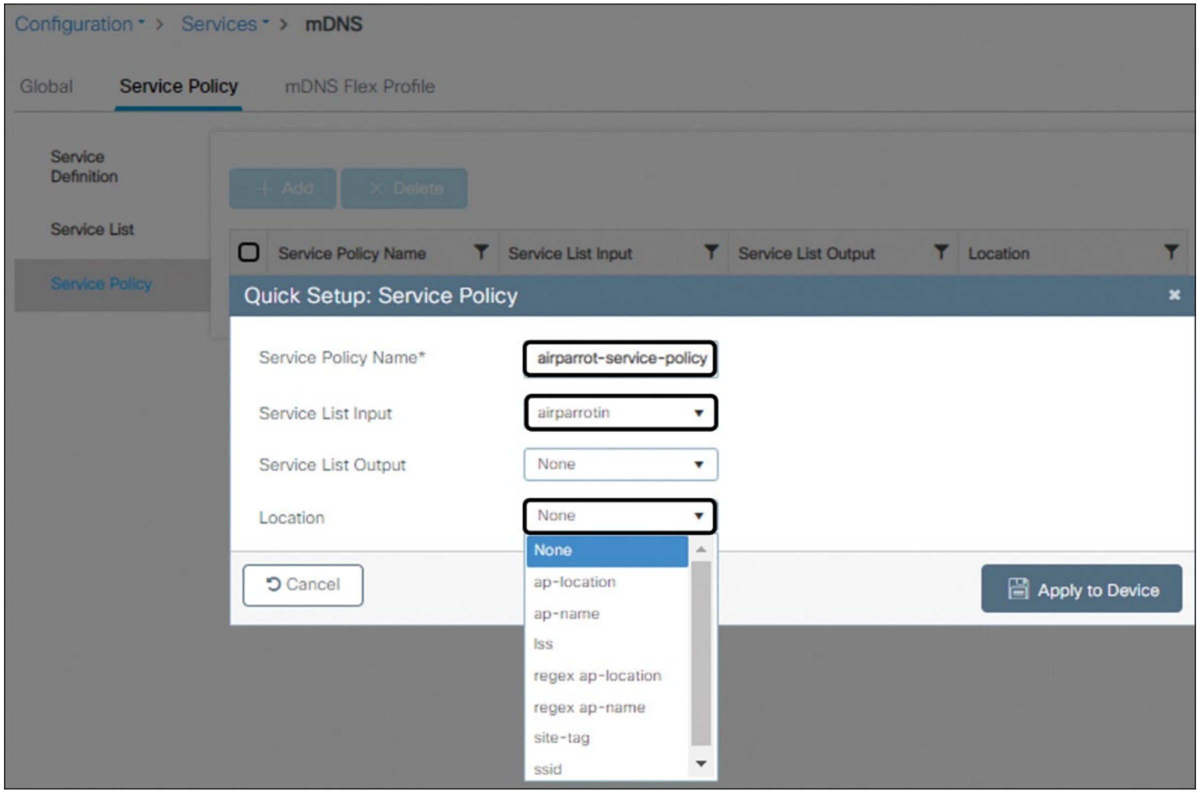


Figure 8-27 Creating a service policy and mapping service filters

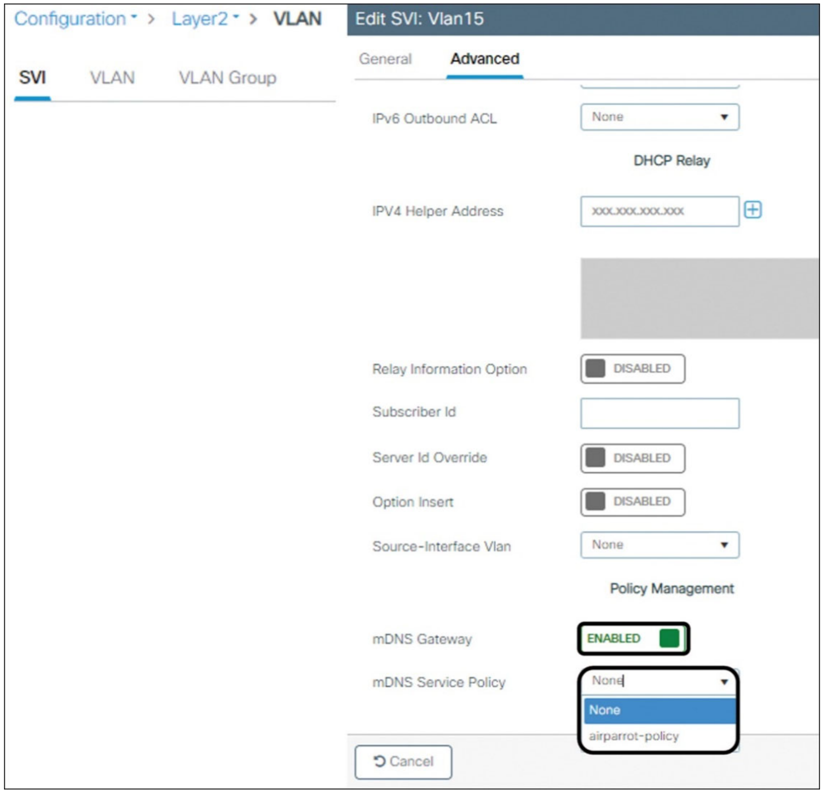


Figure 8-28 Enabling the mDNS gateway on a VLAN SVI

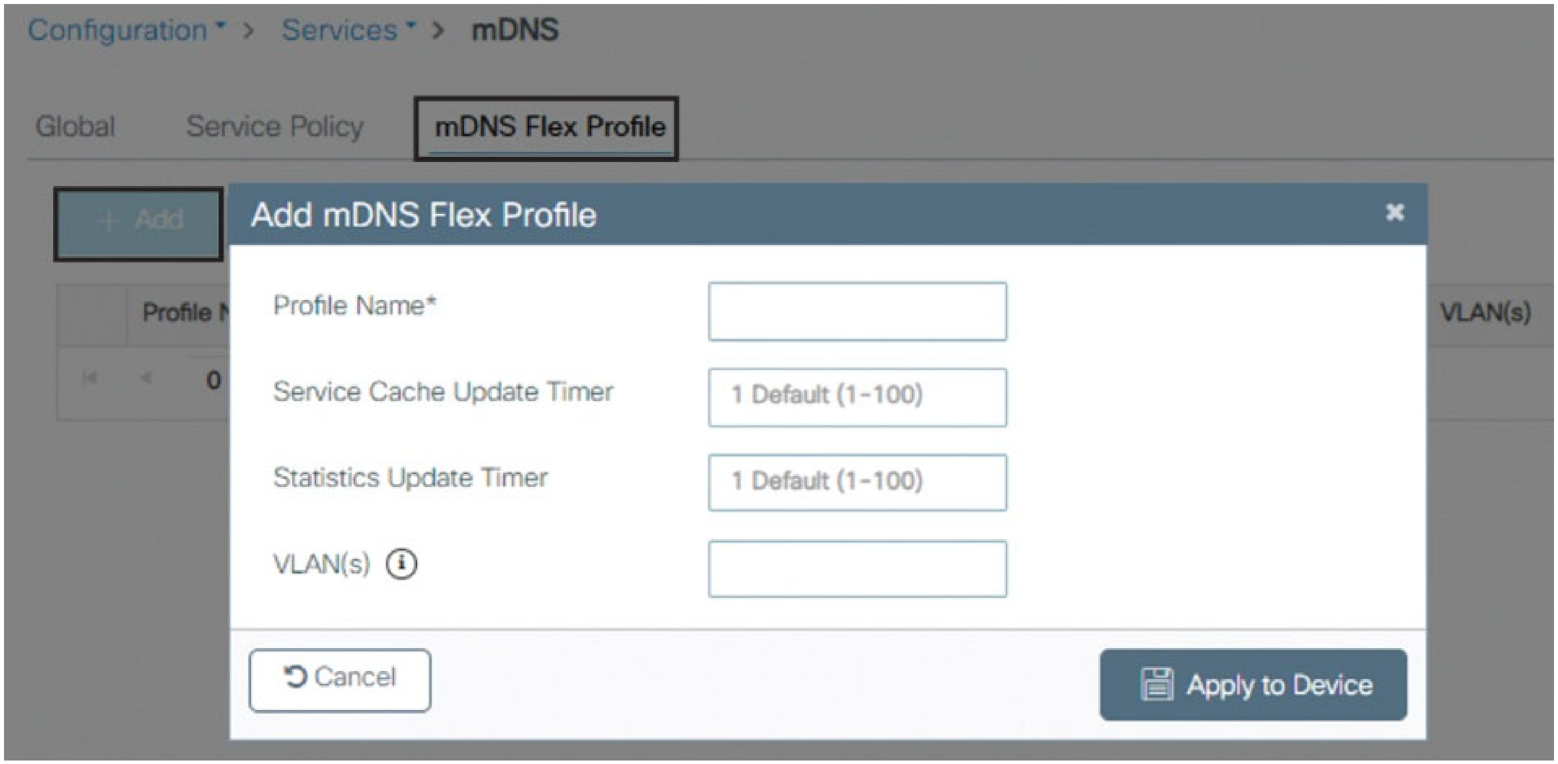


Figure 8-29 Defining an mDNS Flex profile

Figure 9-1 Simplified view on an enterprise network

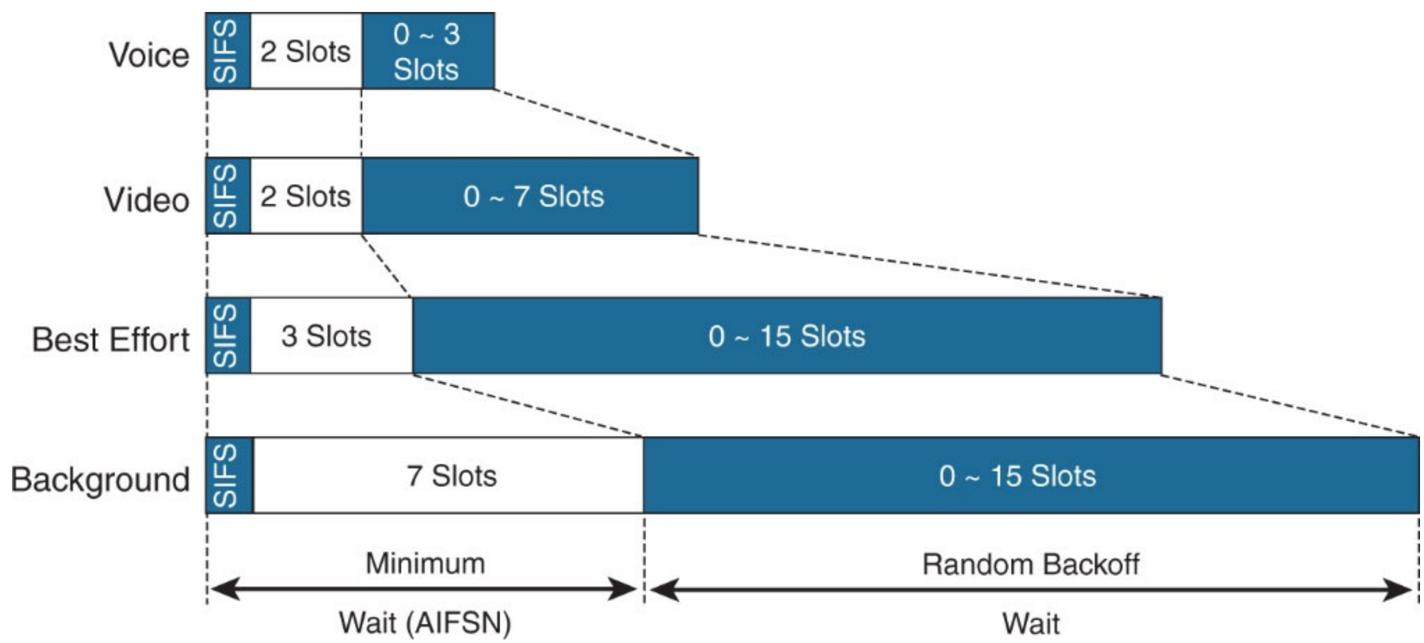


Figure 9-2 EDCA slots for each access category

EDCA/WMM AC	TXOP (µs)	TXOP (Units)
Voice	2080	65
Video	4096	128
Best Effort	2528	79
Background	2528	79

Figure 9-3 TXOP values for each access category

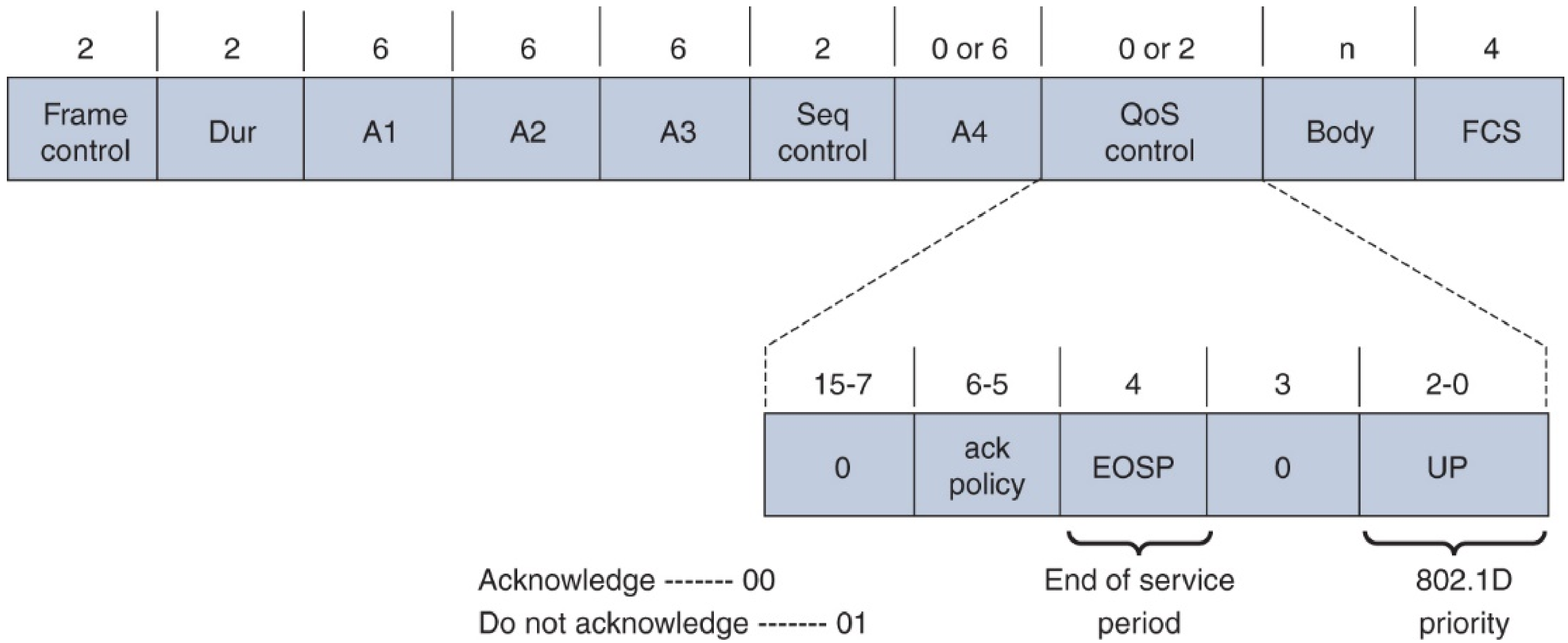


Figure 9-4 The UP (802.1D) bits in the 802.11 header

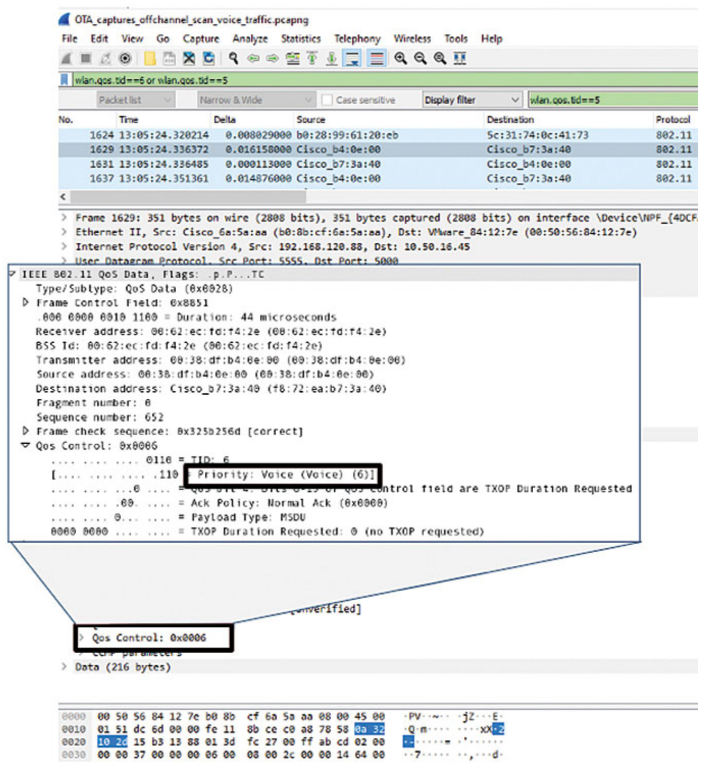


Figure 9-5 The UP value in a real over-the-air capture

Access Category (AC)	UP values
Background (AC_BK)	1,2
Best Effort (AC_BE)	0,3
Video (AC_VI)	4,5
Voice (AC_V0)	6,7

Figure 9-6 Table with UP to Access Categories mapping

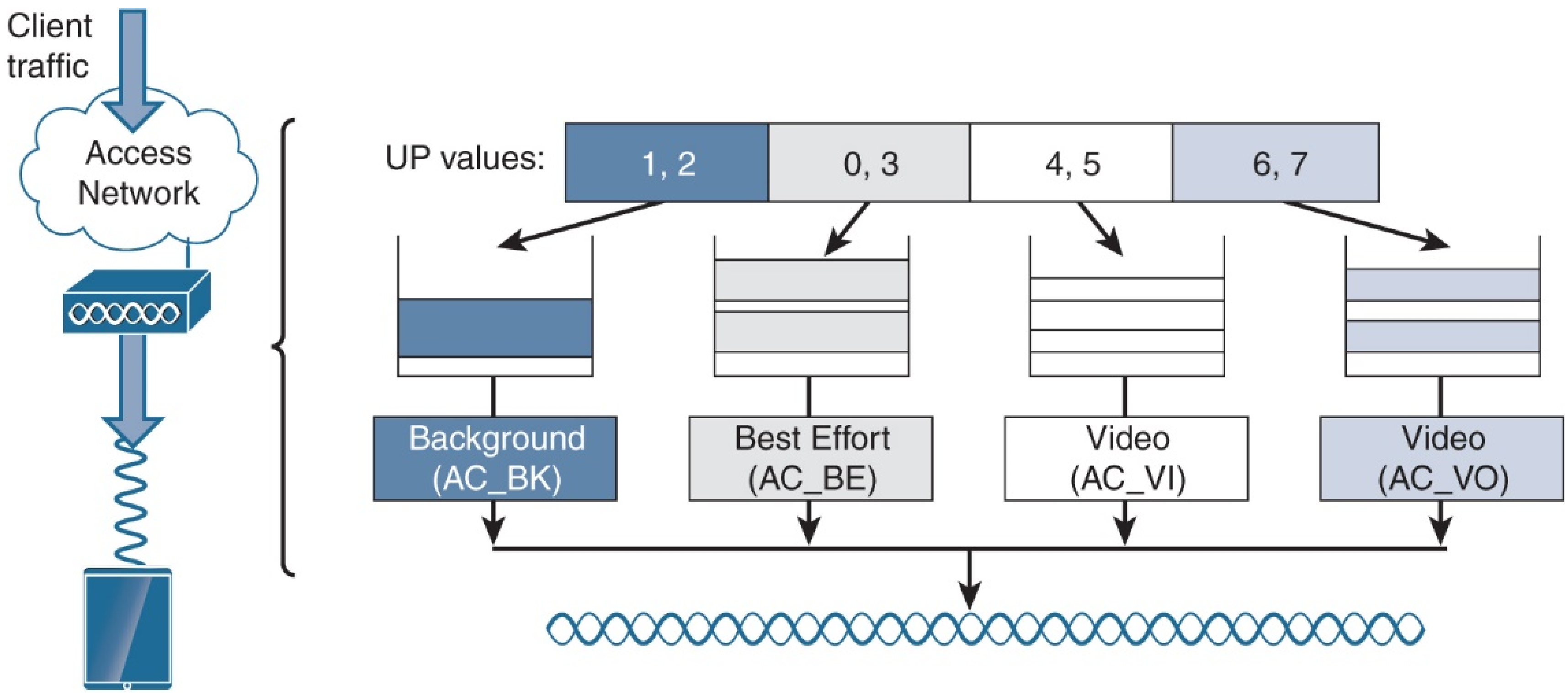


Figure 9-7 Client UP value to queue mapping at the AP

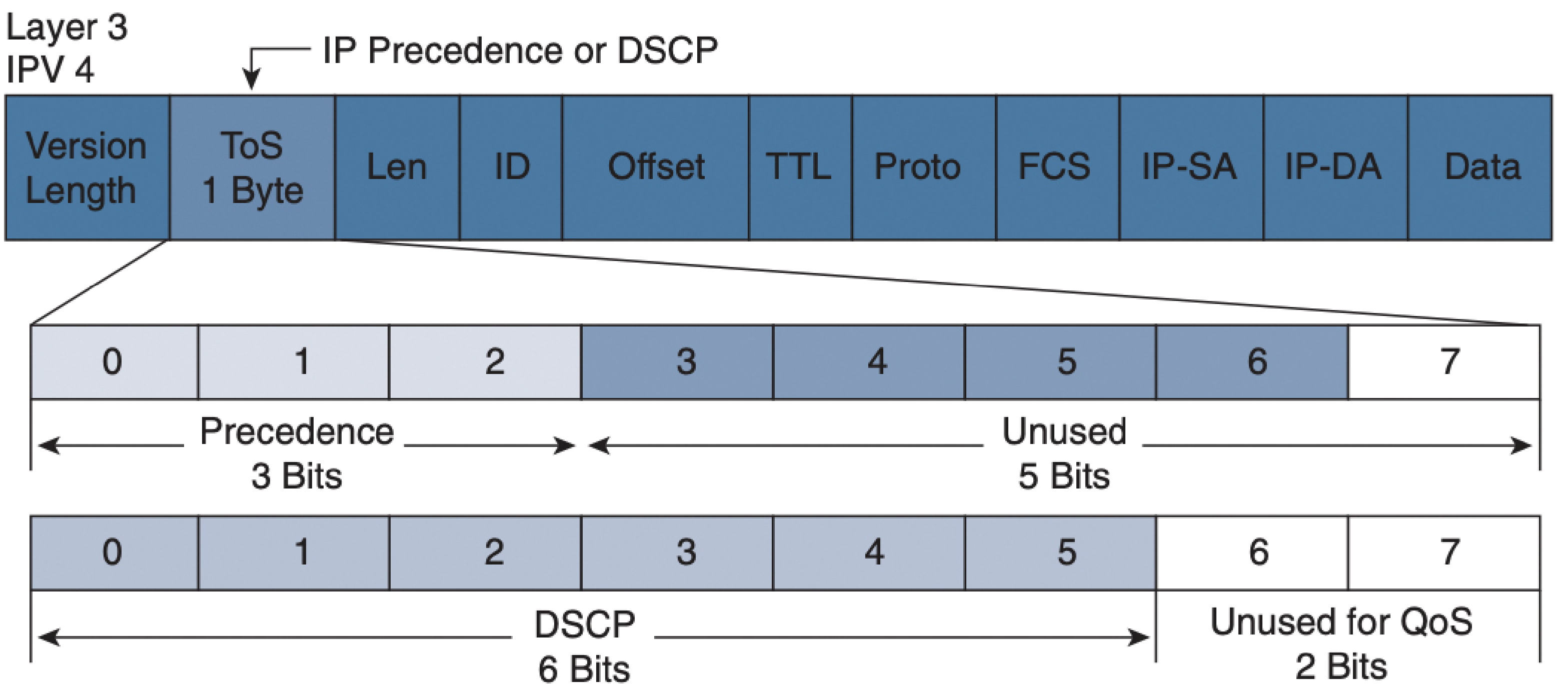
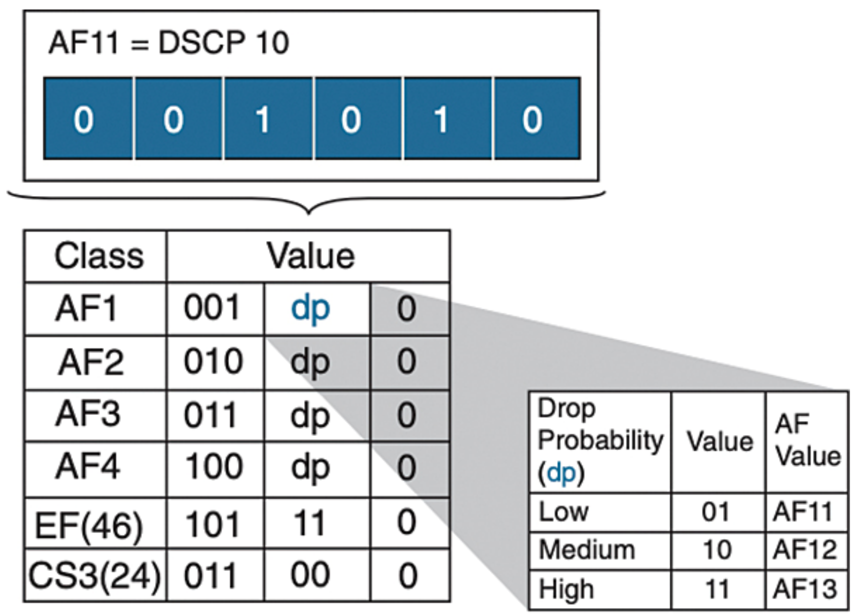


Figure 9-8 DSCP value in the IPv4 header



AF = Assured Forwarding (DSCP 10 to DSCP 38)
EF = Expedited Forwarding (DSCP 46)
CS = Class Selector. Used to preserve partial backward compatibility with IP precedence.

Figure 9-9 Class Selector and DP bits

DSCP Class	DSCP (bin)	DSCP (hex)	DSCP (dec)
none	000000	0x00	0
cs1	001000	0x08	8
af11	001010	0x0A	10
af12	001100	0x0C	12
af13	001110	0x0E	14
cs2	010000	0x10	16
af21	010010	0x12	18
af22	010100	0x14	20
af23	010110	0x16	22
cs3	011000	0x18	24
af31	011010	0x1A	26
af32	011100	0x1C	28
af33	011110	0x1E	30
cs4	100000	0x20	32
af41	100010	0x22	34
af42	100100	0x34	36
af43	100110	0x26	38
cs5	100100	0x28	40
ef	101110	0x2E	46
cs6	110000	0x30	48
cs7	111000	0x38	56

Figure 9-10 DSCP classes and the associated values

RFC 4594-Based Model	DSCP
Network Control	(CS7)
Internetwork Control	CS6
Voice + DSCP-Admit	EF +44
Broadcast Video	CS5
Multimedia Conferencing	AF4
Real-time Interaction	CS4
Multimedia Streaming	AF3
Signaling	CS3
Transactional Data	AF2
OAM	CS2
Bulk Data	AF1
Scavenger	CS1
Best Effort	DF

Figure 9-11 DSCP classes and the associated values

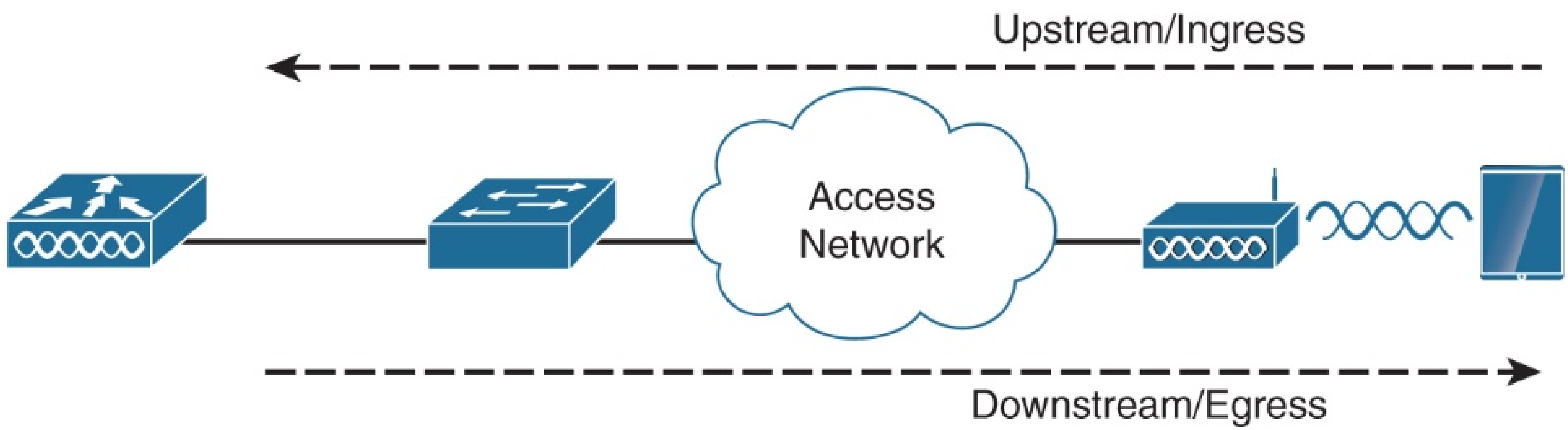


Figure 9-12 Upstream and downstream directions

IETF DiffServ Service Class	DSCP name	DSCP value	UP	Access Category
Network Control (Reserved)	CS7	56	0	AC_BE
Network Control or Internet-work	CS6	48	0	AC_BE
Voice (and Voice-admit)	EF	46, 44	6	AC_VO
Signaling	CS5	40	5	AC_VI
Multimedia Conferencing	AF41,AF42, AF43	34, 36, 38	4	AC_VI
Real-Time Interactive	CS4	32	5	AC_VI
Multimedia Streaming	AF31, AF32, AF33	26, 28, 30	4	AC_VI
Broadcast Video	CS3	24	4	AC_VI
Low-Latency Data	AF21, AF22, AF23	18, 20, 22	3	AC_BE
OAM	CS2	16	0	AC_BE
High-Throughput Data	AF11, AF12, AF13	10, 12, 14	2	AC_BK
Standard	DF (default forwarding)	0	0	AC_BE
Low-Priority Data	CS1	8	1	AC_BK

Figure 9-13 DSCP to UP mapping based on RFC 8325

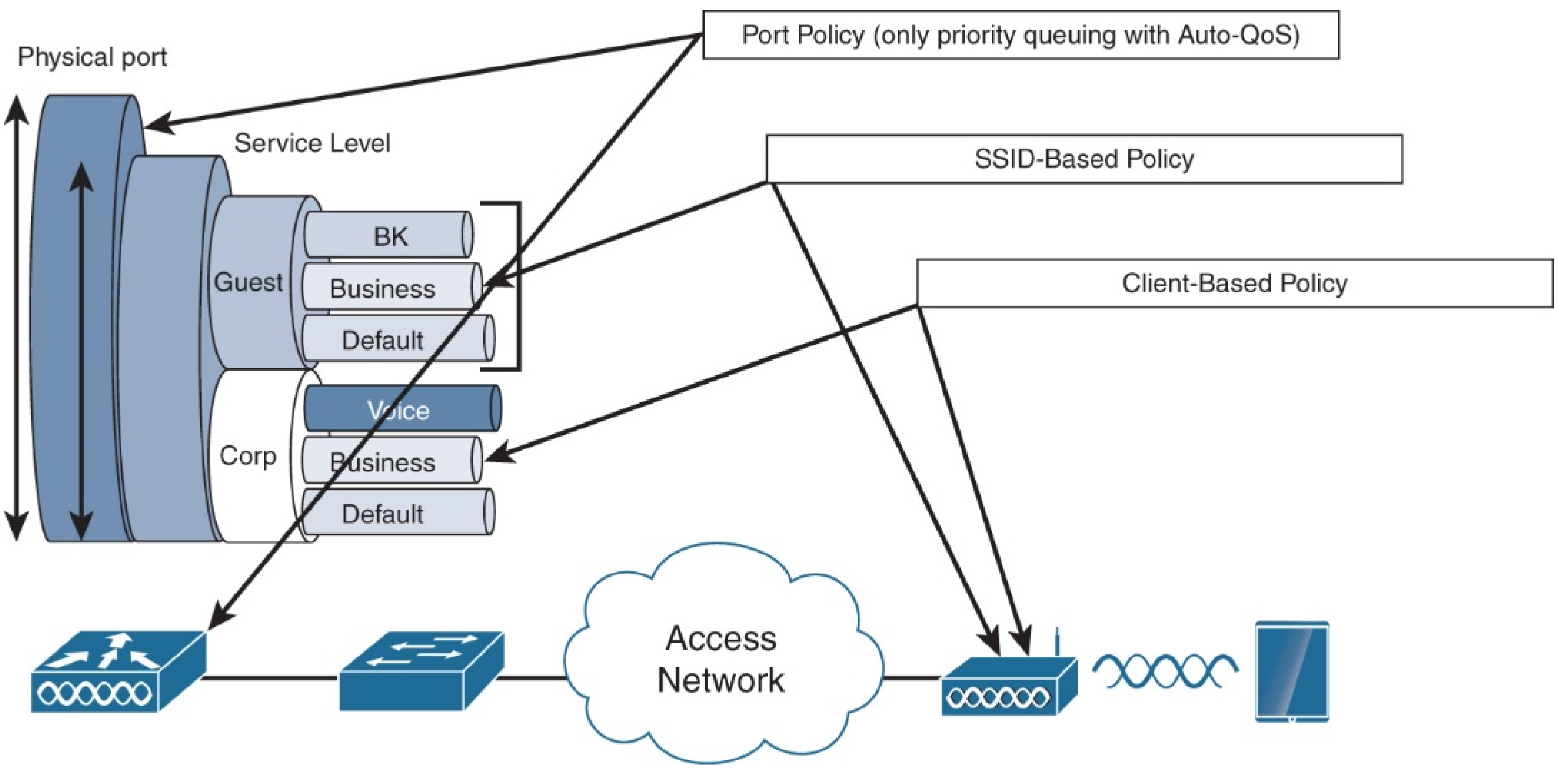


Figure 9-14 QoS policy targets

Classification ACL

```
ip access-list extended AutoQos-4.0-Output-Acl-CAPWAP-C
 10 permit udp any eq 5246 16666 any
```

Class-map definition

```
class-map match-any AutoQos-4.0-Output-CAPWAP-C-Class
 match access-group name AutoQos-4.0-Output-Acl-CAPWAP-C
class-map match-any AutoQos-4.0-Output-Voice-Class
 match dscp ef
```

Policy-map definition

```
policy-map AutoQos-4.0-wlan-Port-Output-Policy
 class AutoQos-4.0-Output-CAPWAP-C-Class
  priority level 1
 class AutoQos-4.0-Output-Voice-Class
  priority level 2
 class class-default
```

Service-policy attachment

```
interface TenGigabitEthernet0/0/0
 service-policy output AutoQos-4.0-wlan-Port-Output-Policy
```

Figure 9-15 MQC policy

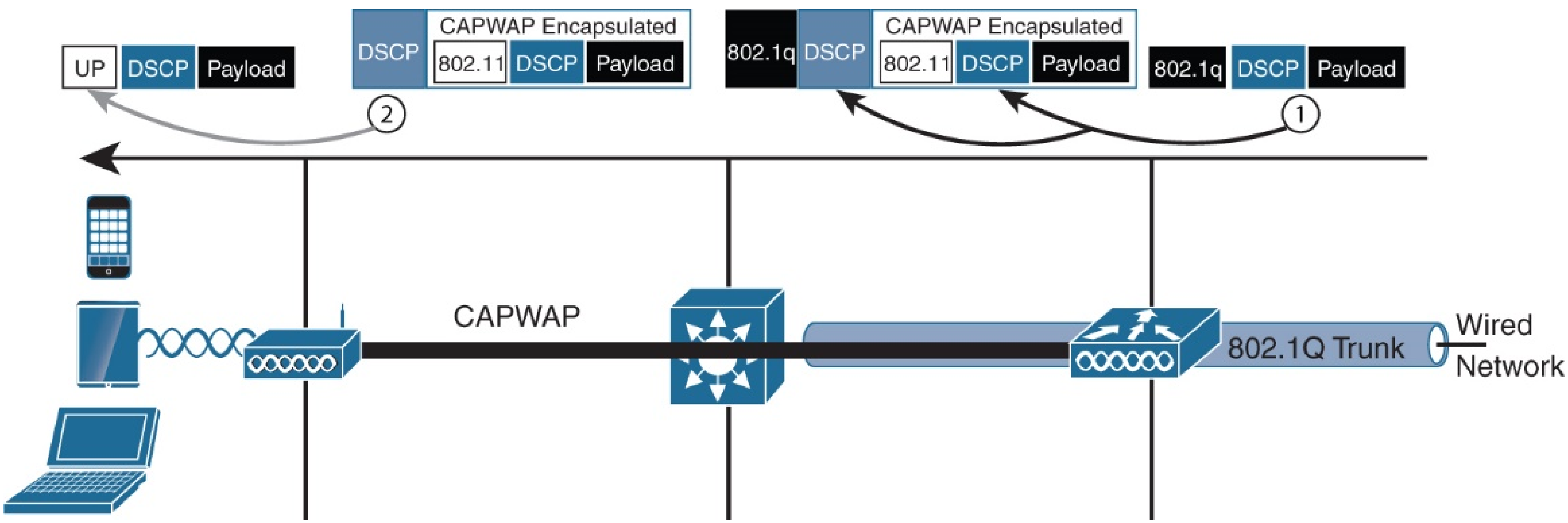


Figure 9-16 “Trust” DSCP in the downstream direction

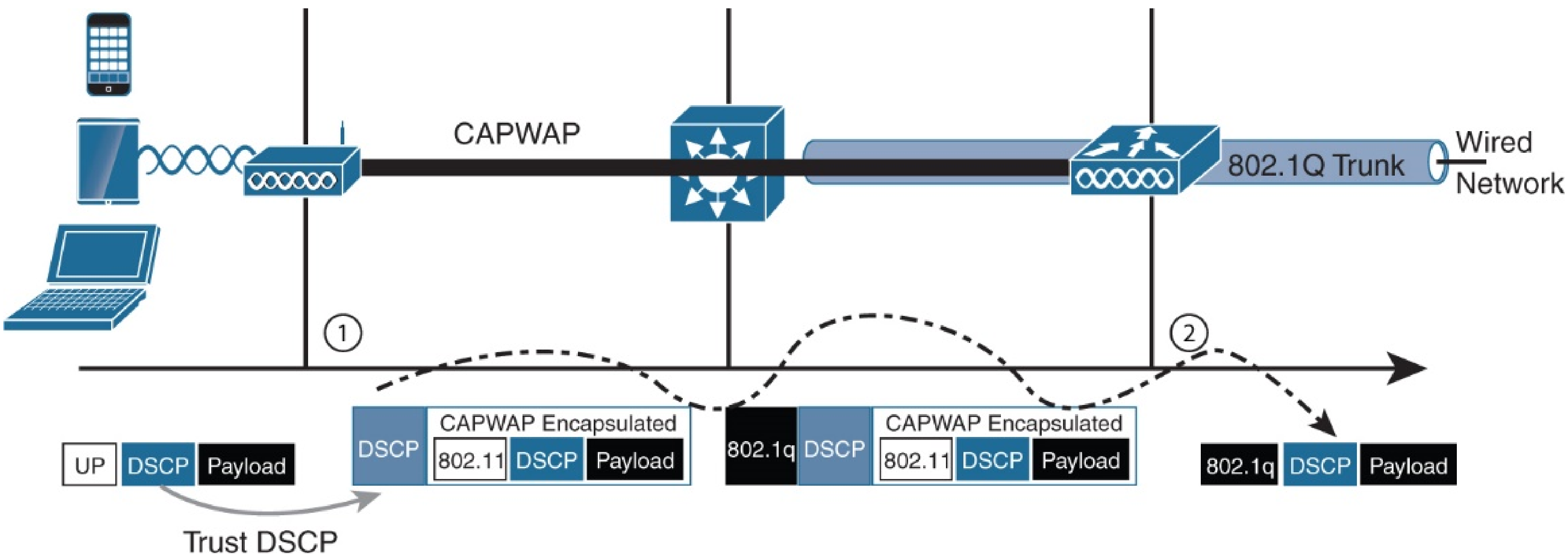


Figure 9-17 “Trust” DSCP in the upstream direction

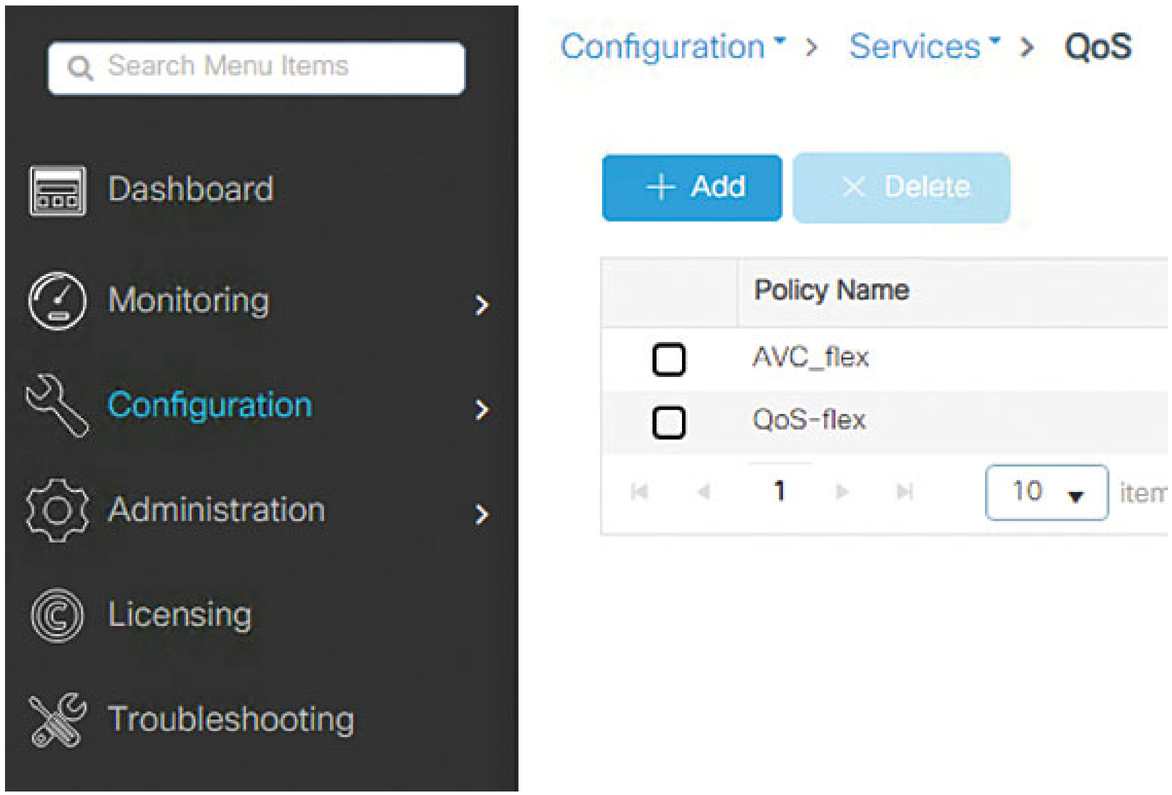


Figure 9-18 Click Add to start configuring a new QoS policy

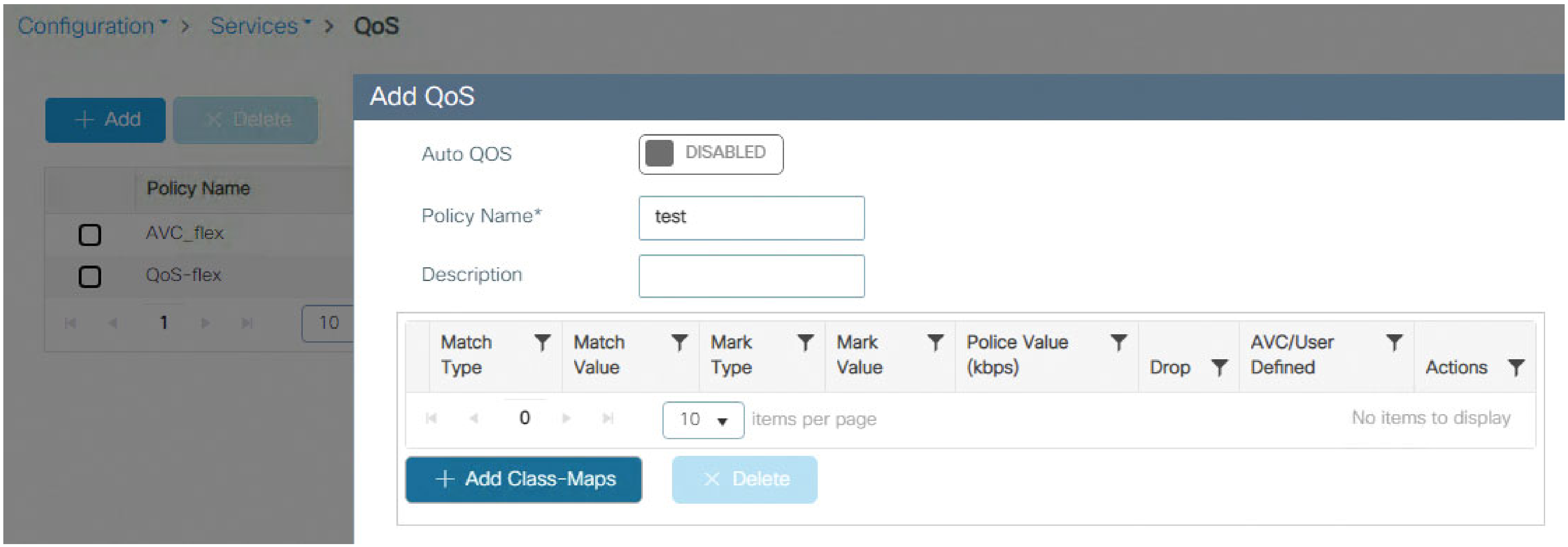


Figure 9-19 Start adding the classification logic

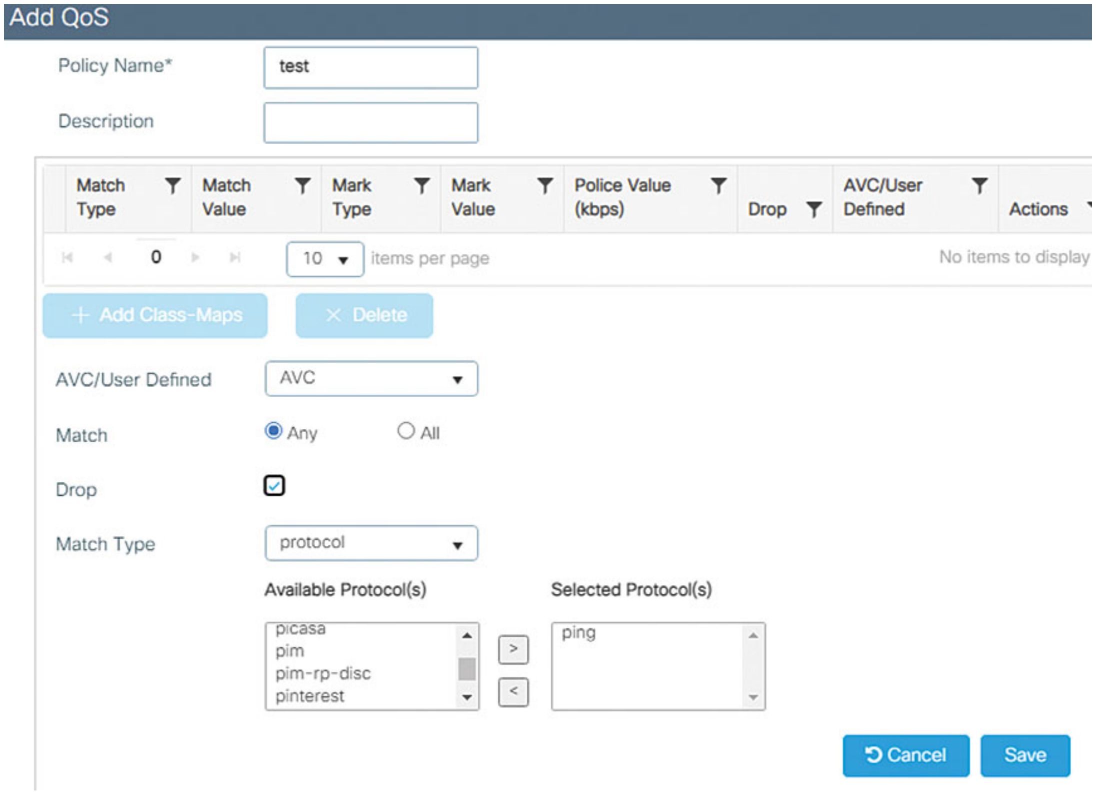


Figure 9-20 Configuring a policy to match and drop ping traffic

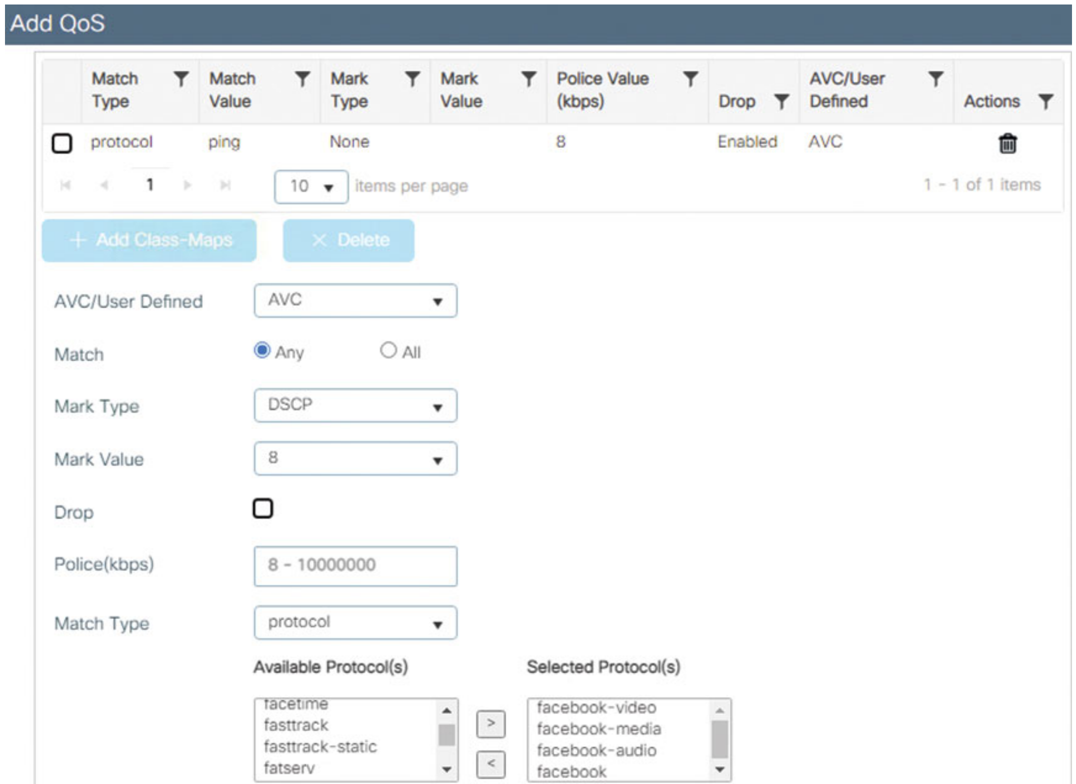


Figure 9-21 Configuring a policy to match and mark down Facebook traffic

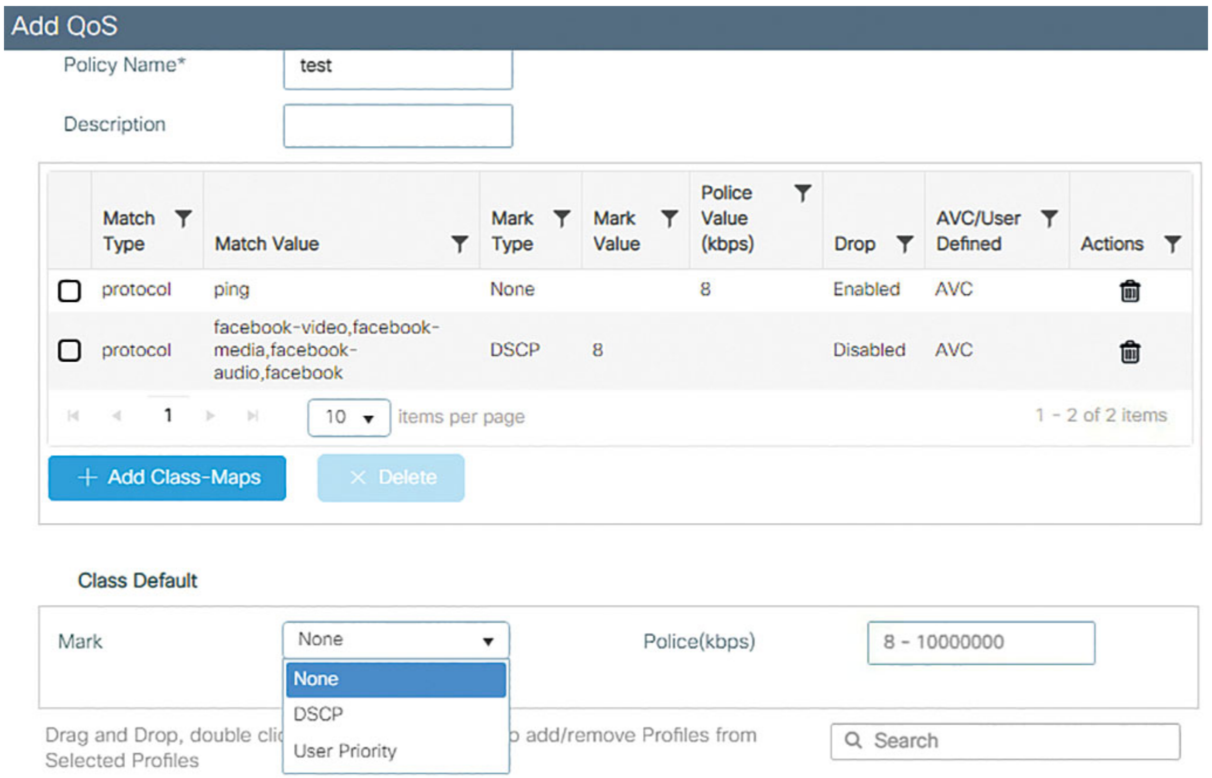


Figure 9-22 Configuring a policy for class-default

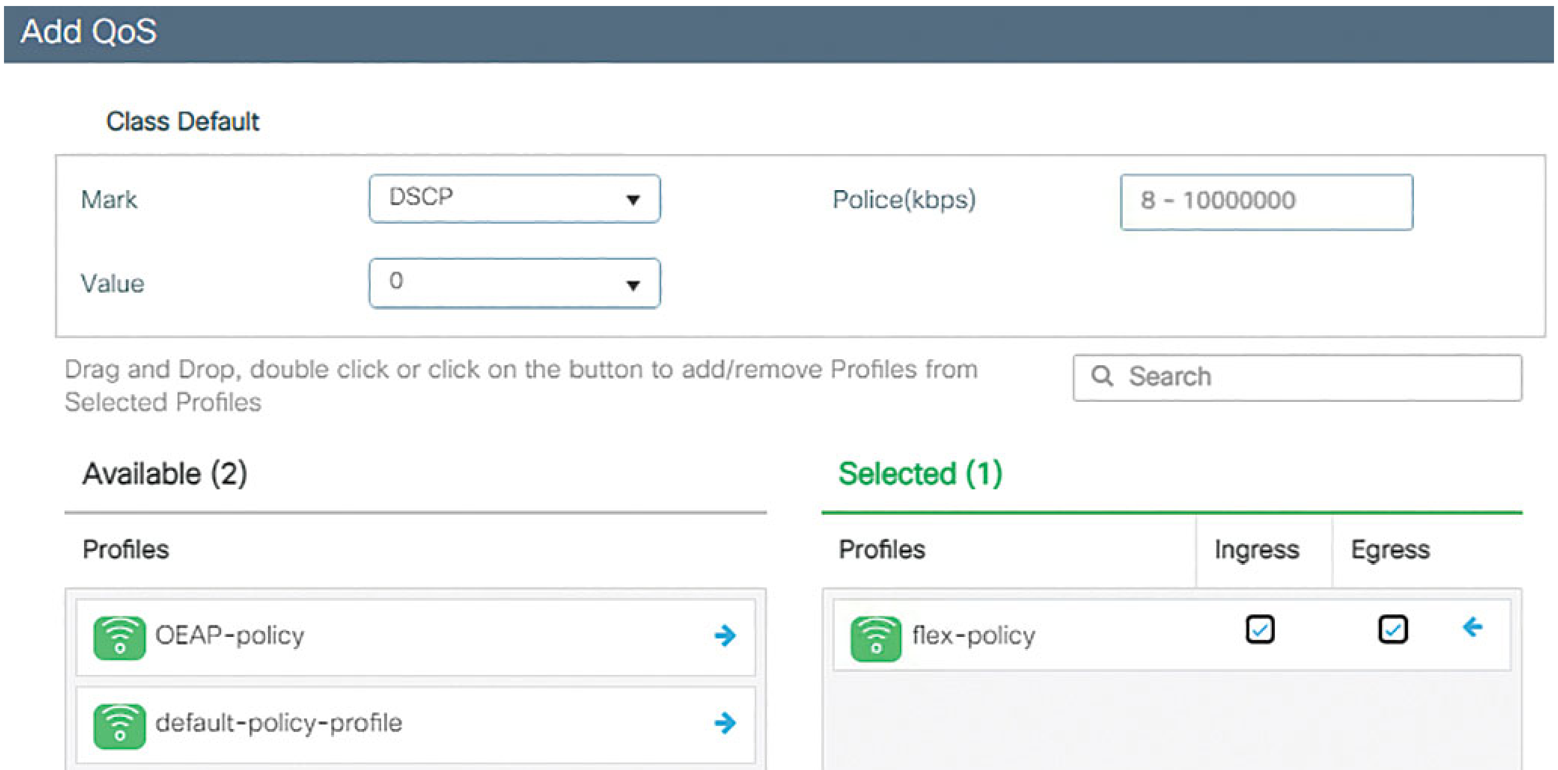


Figure 9-23 Assigning the policy to the policy profile(s)

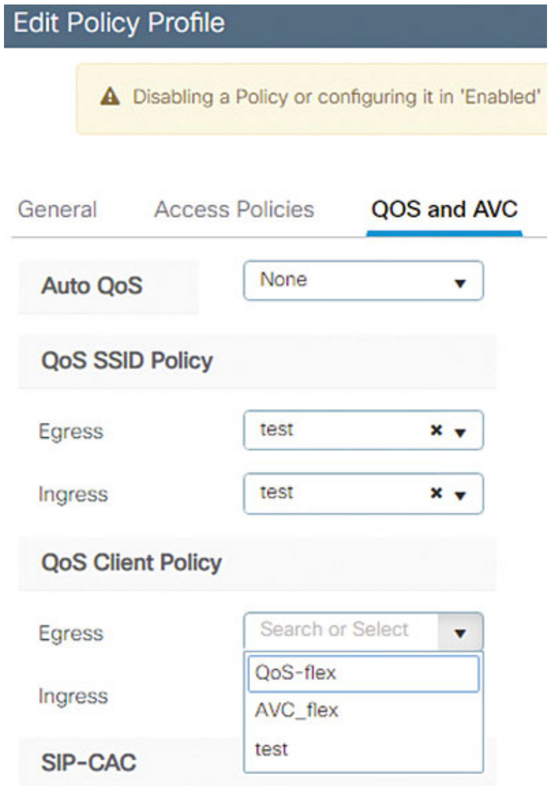


Figure 9-24 Assigning the policy at SSID or client level, or both

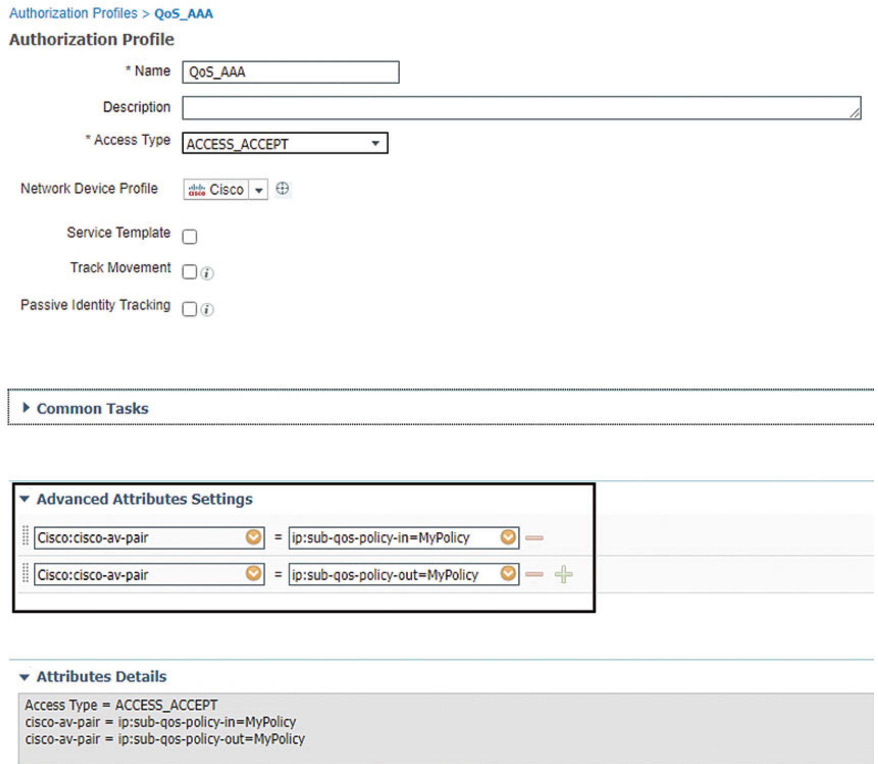


Figure 9-25 Authorization profile in ISE for AAA QoS override

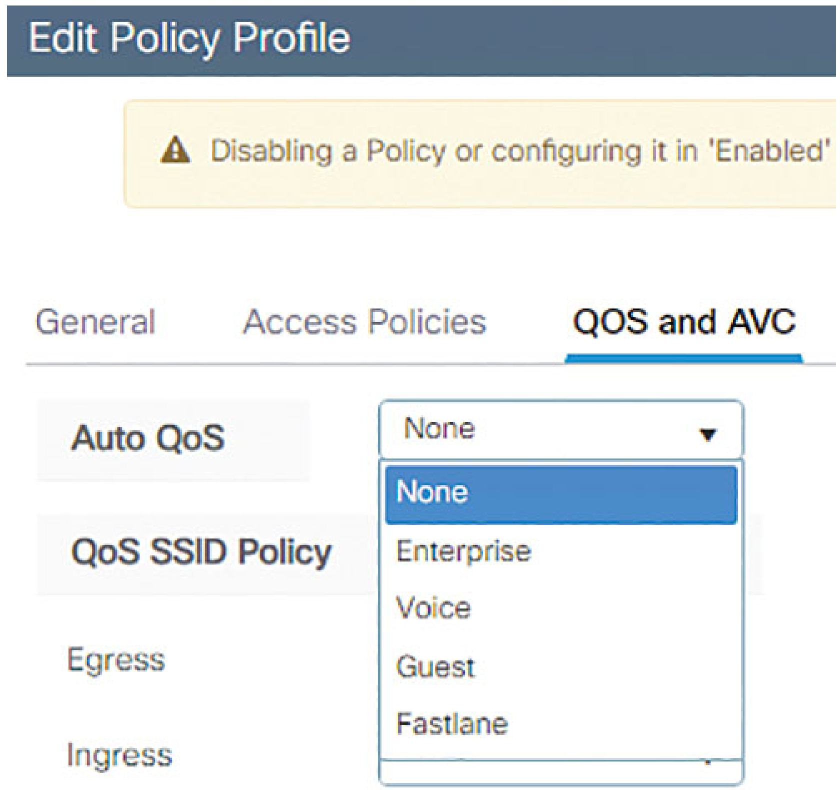


Figure 9-26 Auto QoS configuration under the Policy profile

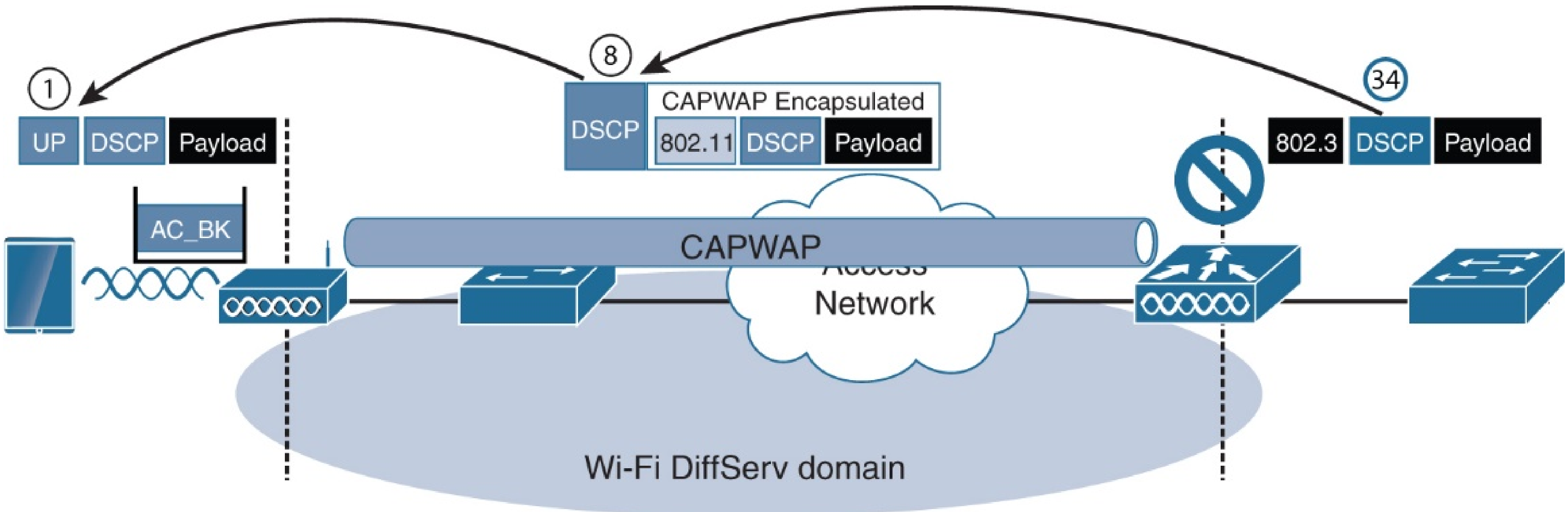


Figure 9-27 Bronze QoS profile, downstream

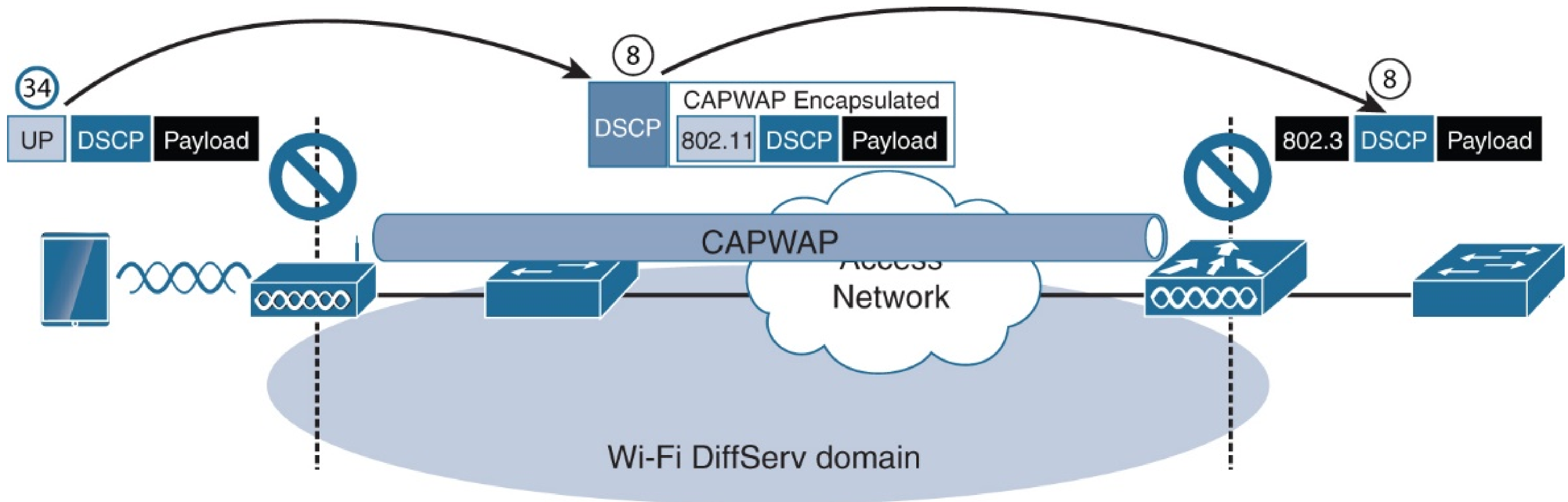


Figure 9-28 Bronze QoS profile, upstream

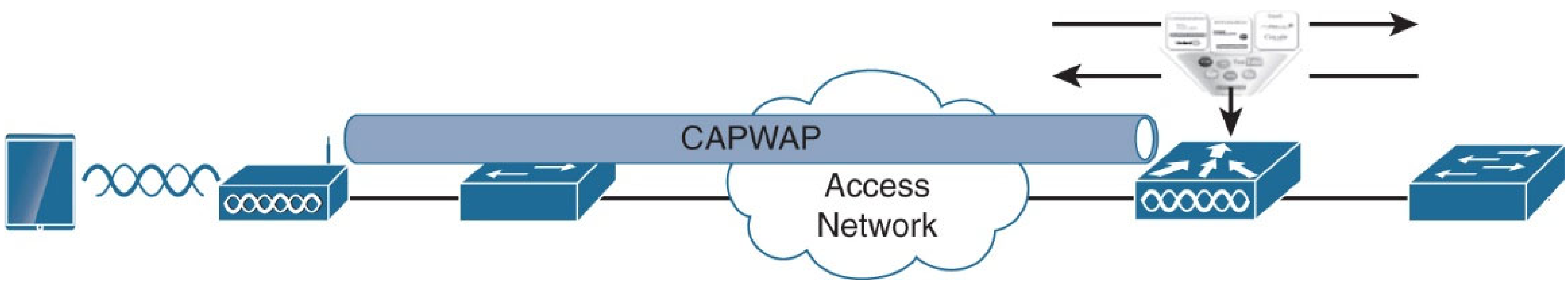


Figure 9-29 AVC in local mode and Flex central switching is applied at the WLC

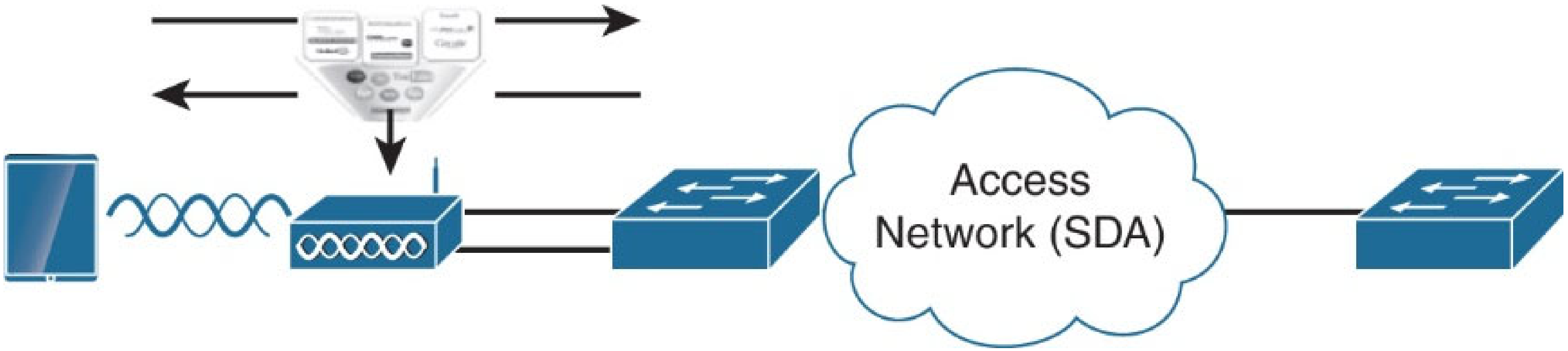


Figure 9-30 AVC in FlexConnect local switching and SDA wireless is applied at the AP

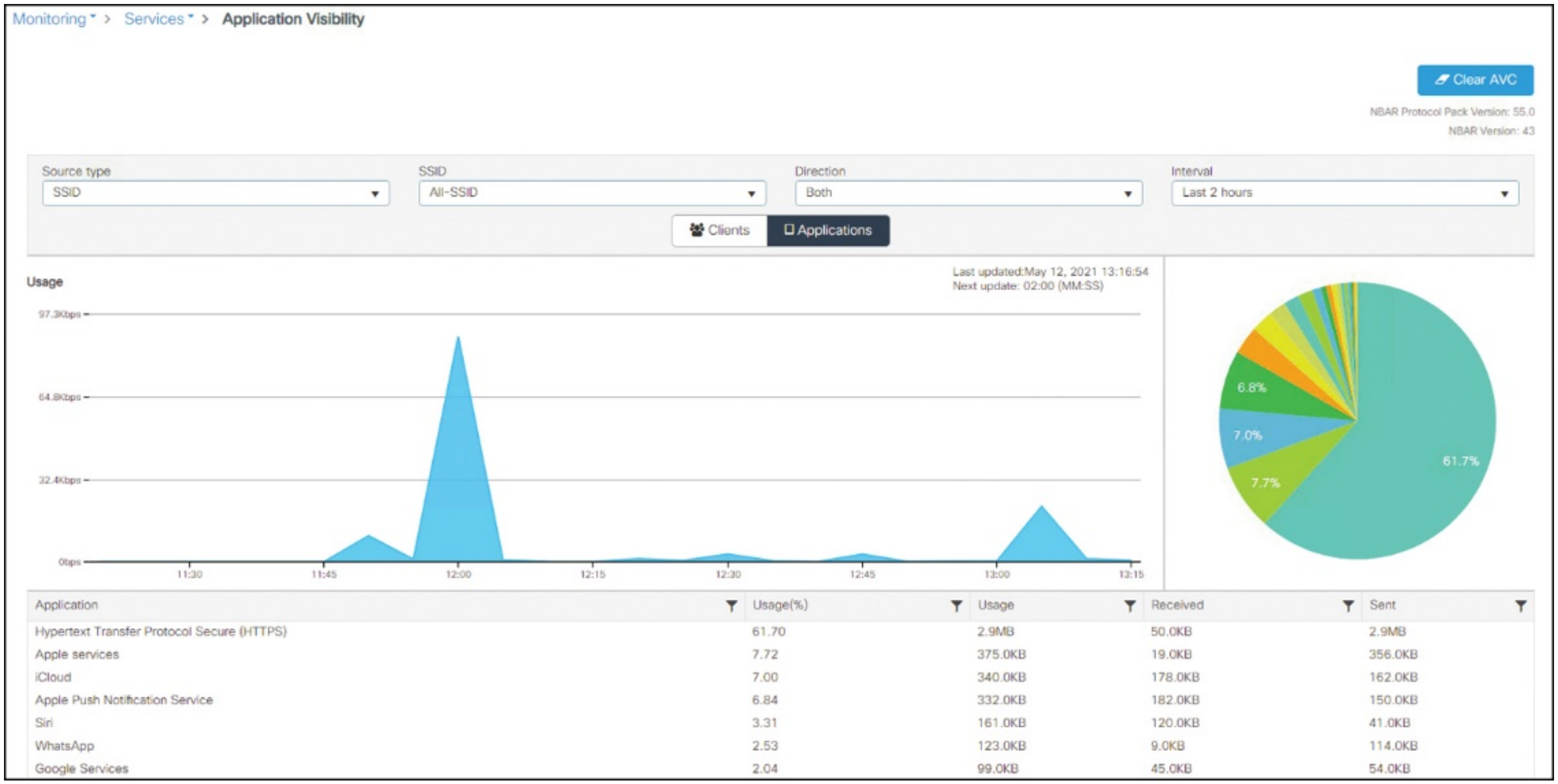


Figure 9-31 Monitoring AVC

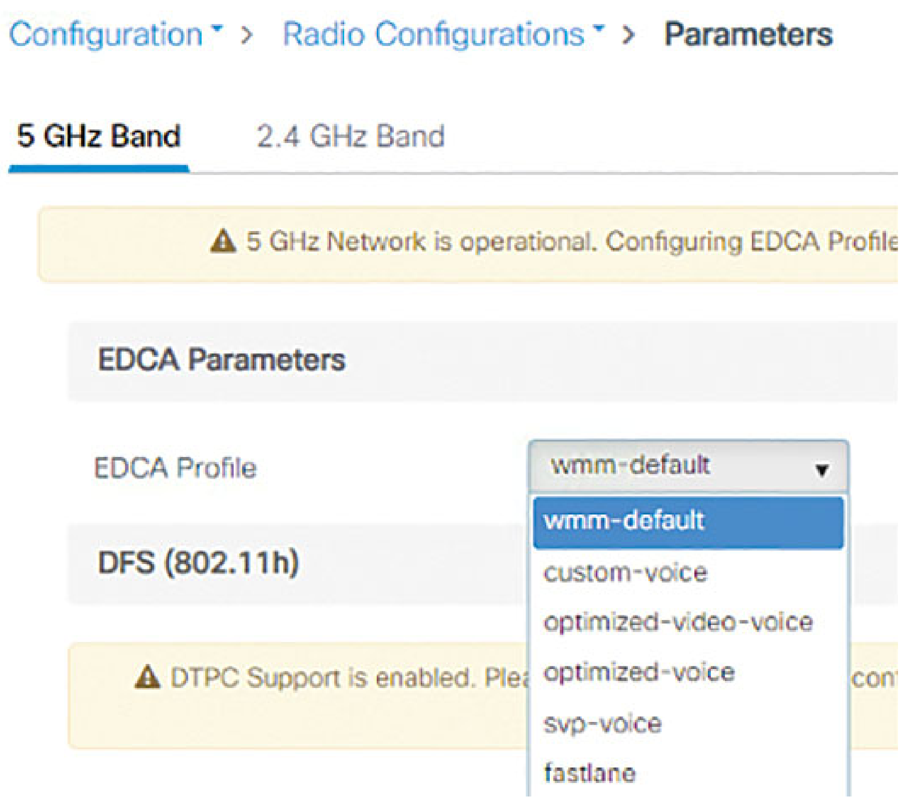


Figure 9-32 EDCA profile settings for a 5 GHz network

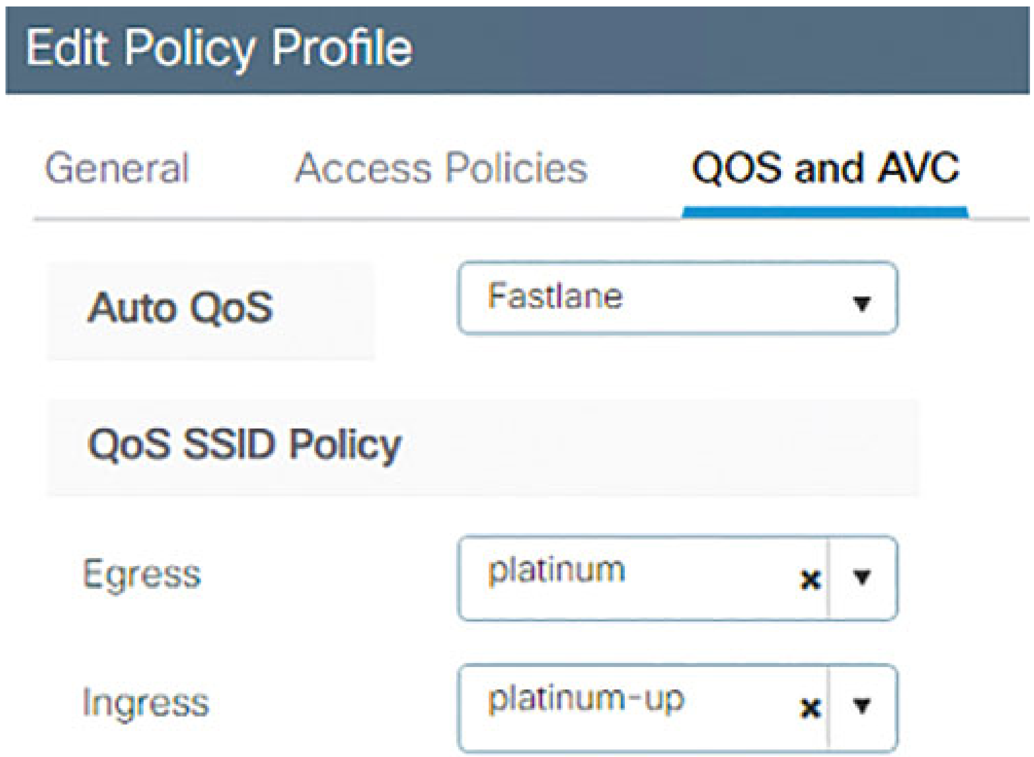


Figure 9-33 Recommended policy profile settings for an SSID with voice traffic

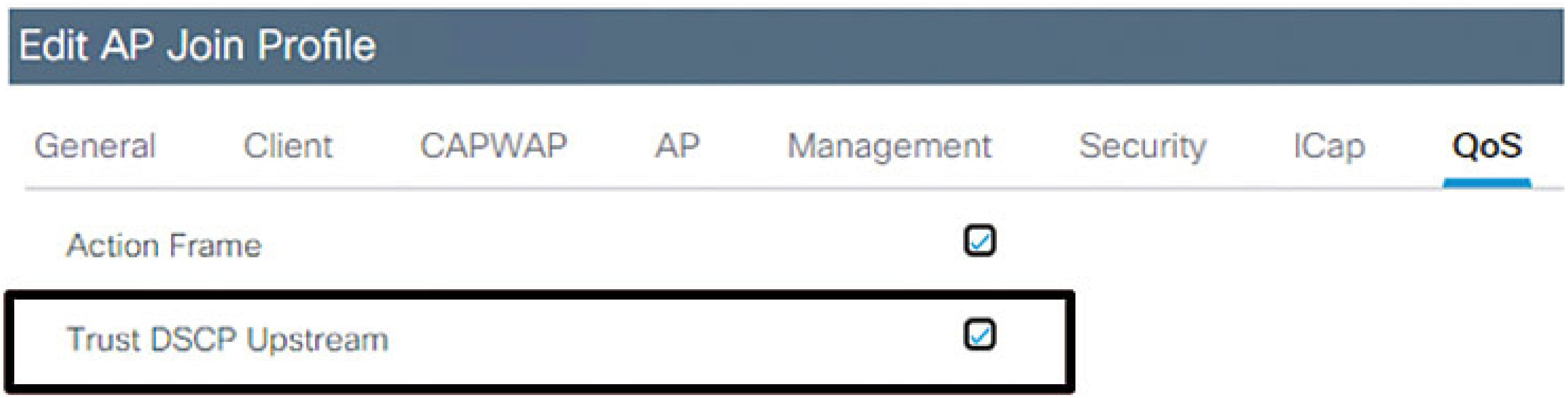


Figure 9-34 Trust DSCP settings under the AP Join Profile

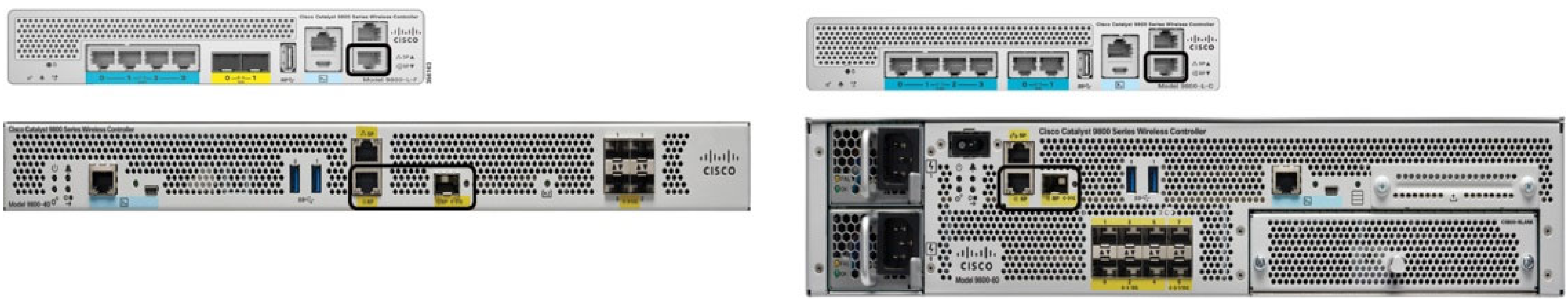


Figure 10-1 Redundancy port on the C9800L-C, C9800L-F, C9800-40, and C9800-80

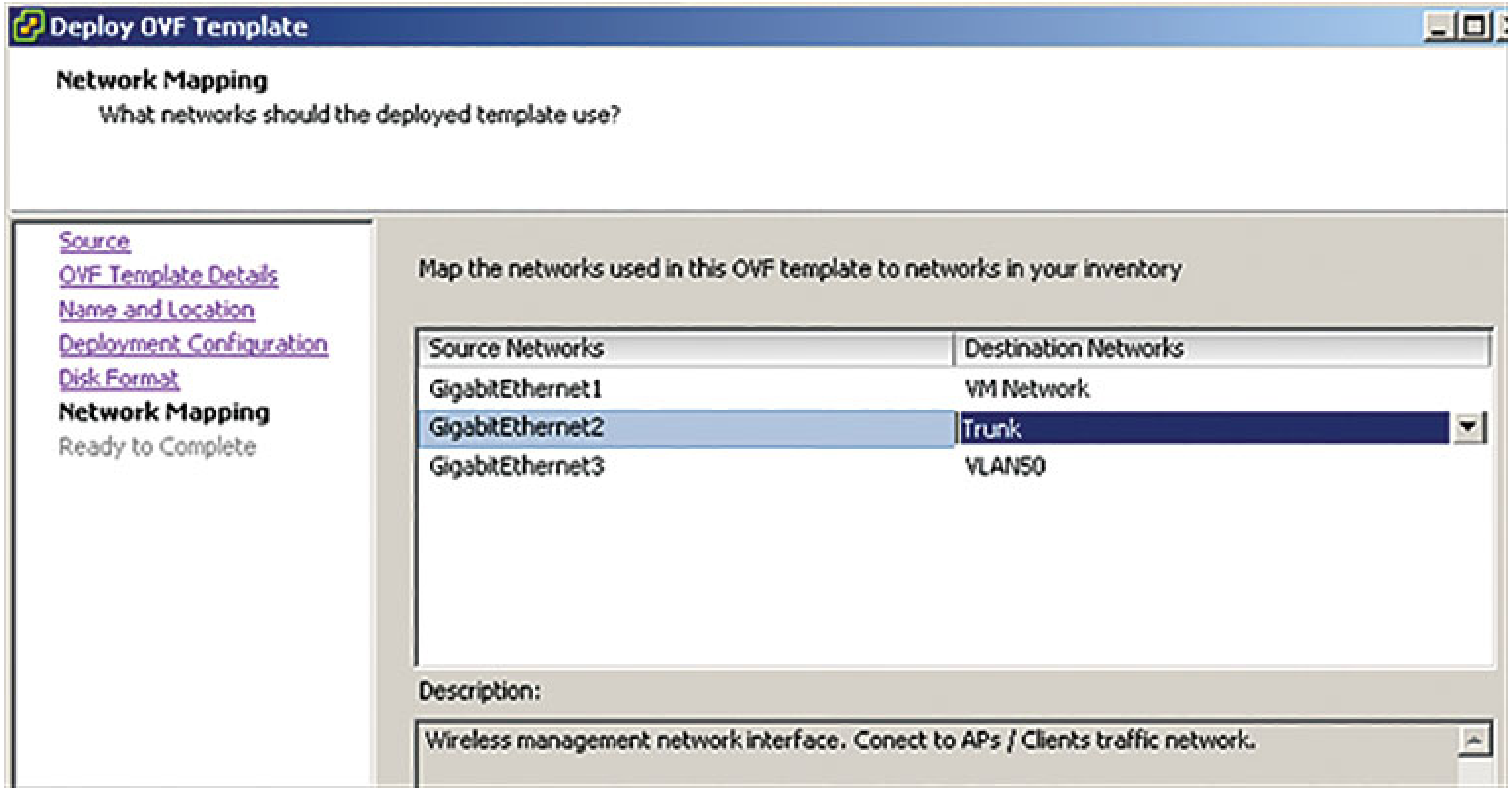


Figure 10-2 vNIC mapping for a C9800-CL over ESXi

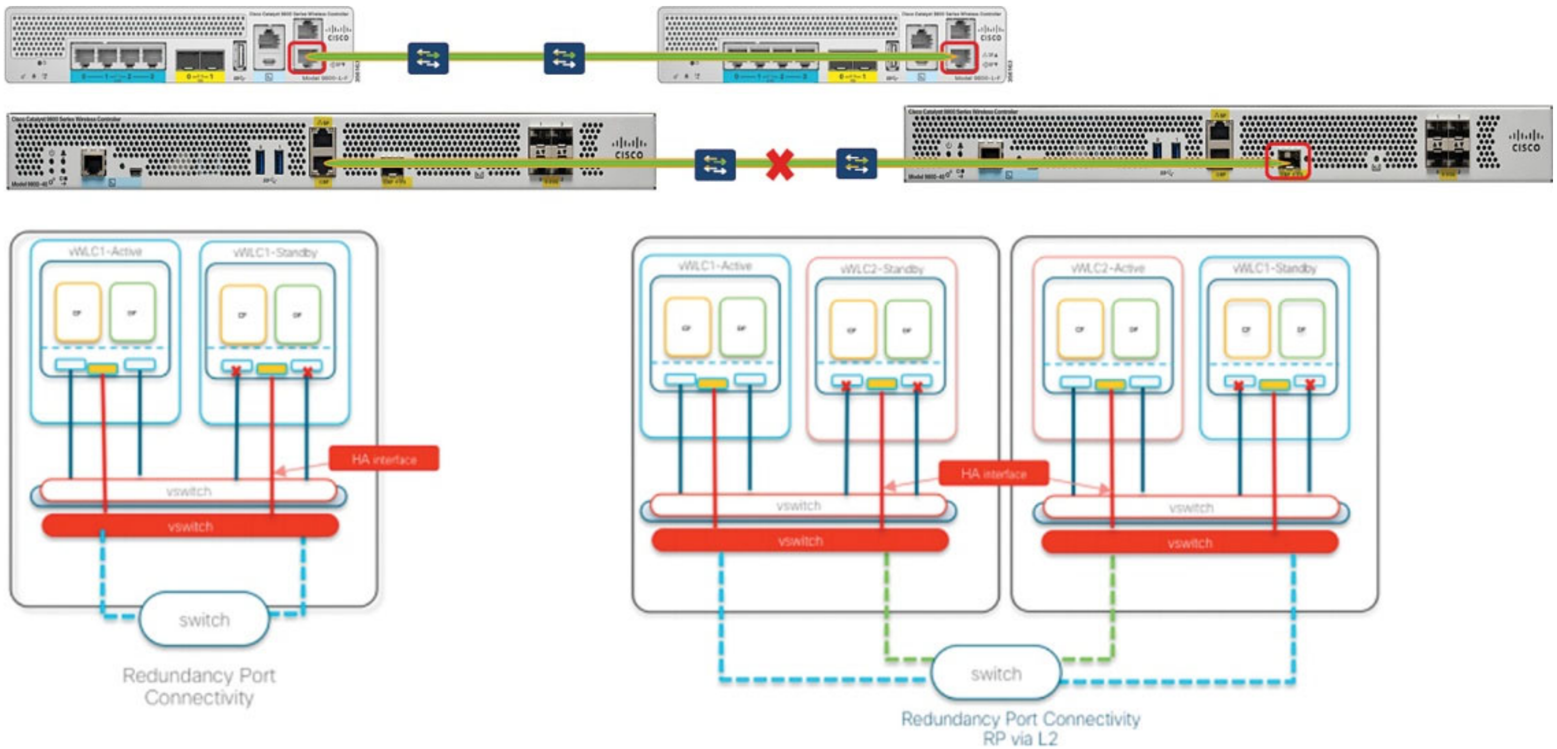


Figure 10-3 RP-to-RP connectivity between two C9800-L-Cs, two C9800-40s, and two C9800-CLs

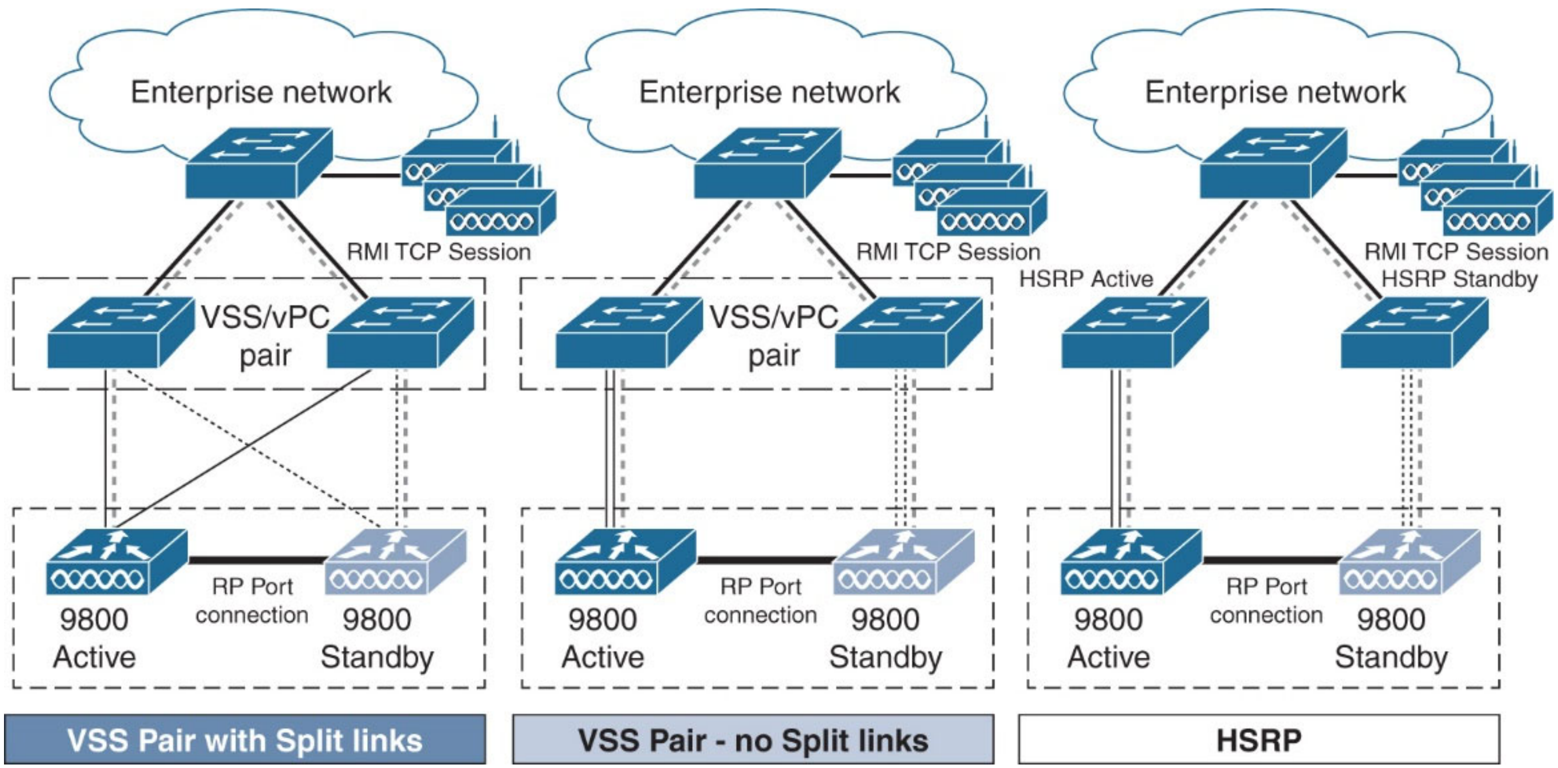


Figure 10-4 Supported topologies from an RP+RMI C9800 HA pair

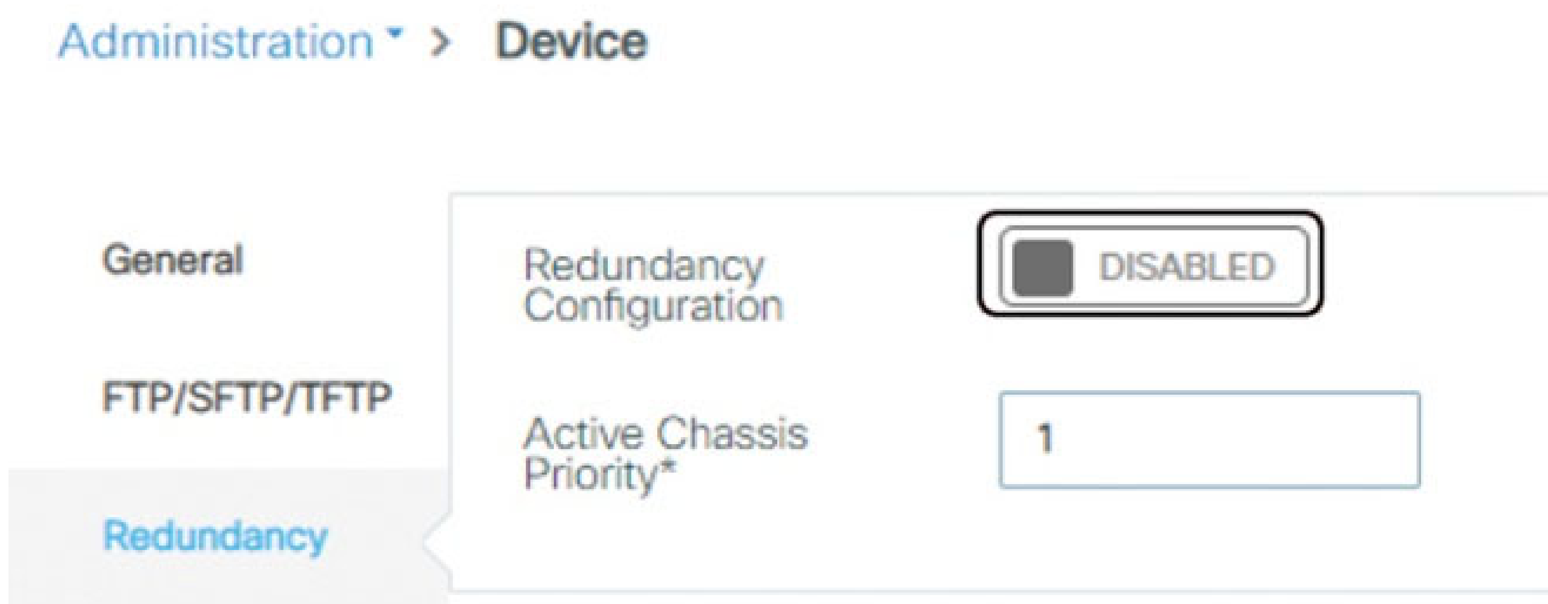


Figure 10-5 Enabling the HA SSO

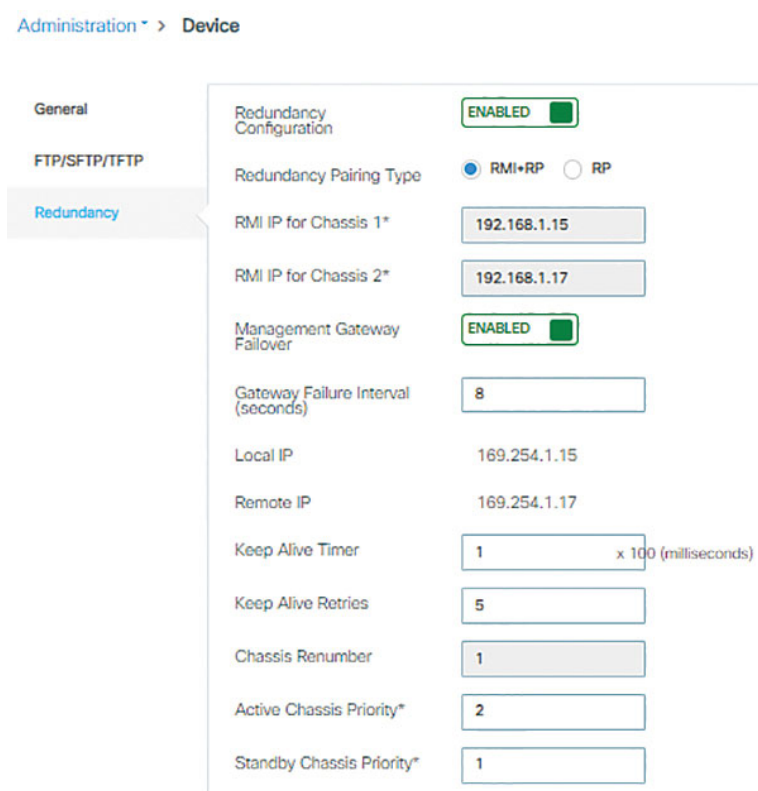


Figure 10-6 Configuring SSO redundancy on a C9800

```
C9800-L-X-K9 platform with 16777216 Kbytes of main memory

File size is 0x000015cf
Located packages.conf
Image size 5583 inode num 26, bks cnt 2 blk size 8*512
#
File size is 0x023f44c5
Located C9800-L-rpboot.17.06.01.SPA.pkg
Image size 37700805 inode num 506916, bks cnt 9205 blk size 8*512
###############################################################################################
###############################################################################################
###############################################################################################
########################################################
Boot image size = 37700805 (0x23f44c5) bytes

ROM:RSA Self Test Passed
ROM:Sha512 Self Test Passed

Package header rev 3 structure detected
Calculating SHA-1 hash...done
validate_package_cs: SHA-1 hash:
    calculated 2f5b2f34:80f4af8a:b3b586c1:d41ca412:7736d66f
    expected   2f5b2f34:80f4af8a:b3b586c1:d41ca412:7736d66f
Validating main package signatures

RSA Signed RELEASE Image Signature Verification Successful.
Image validated
Aug 29 20:43:36.605: %PMAN-3-PROC_EMPTY_EXEC_FILE: R0/0: pvp: Empty executable used for process
bt_logger

Waiting for remote chassis to join
```

Figure 10-7 Chassis initialization before HA pairing

```
*Aug 29 15:10:59.998: %STACKMGR-6-STACK_LINK_CHANGE: Chassis 2 R0/0: stack_mgr: Stack port 1 on
Chassis 2 is up
*Aug 29 15:10:59.998: %STACKMGR-6-STACK_LINK_CHANGE: Chassis 2 R0/0: stack_mgr: Stack port 2 on
Chassis 2 is up
*Aug 29 15:11:00.191: %STACKMGR-6-CHASSIS_ADDED: Chassis 2 R0/0: stack_mgr: Chassis 2 has been added
to the stack.
*Aug 29 15:11:01.468: %PMAN-3-PROC_EMPTY_EXEC_FILE: Chassis 2 R0/0: pvp: Empty executable used for
process bt_logger
*Aug 29 15:11:01.497: %STACKMGR-6-CHASSIS_ADDED: Chassis 2 R0/0: stack_mgr: Chassis 2 has been added
to the stack.
*Aug 29 15:11:03.409: %PMAN-3-PROC_EMPTY_EXEC_FILE: Chassis 2 R0/0: pvp: Empty executable used for
process bt_logger
*Aug 29 15:11:03.497: %STACKMGR-6-CHASSIS_ADDED: Chassis 2 R0/0: stack_mgr: Chassis 2 has been added
to the stack.
*Aug 29 15:11:03.746: %STACKMGR-6-ACTIVE_ELECTED: Chassis 2 R0/0: stack_mgr: Chassis 1 has been
elected ACTIVE.
*Aug 29 15:11:04.390: %PMAN-3-PROC_EMPTY_EXEC_FILE: Chassis 2 R0/0: pvp: Empty executable used for
process bt_logger
*Aug 29 15:11:04.987: %RIF_MGR_FSM-6-RP_LINK_UP: Chassis 2 R0/0: rif_mgr: The RP link is UP.
*Aug 29 15:11:04.987: %STACKMGR-1-DUAL_ACTIVE_CFG_MSG: Chassis 2 R0/0: stack_mgr: Dual Active
Detection link is available now
```

Figure 10-8 Active election

```
*Aug 29 20:44:57.925: %RIF_MGR_FSM-6-GW_REACHABLE_ACTIVE: Chassis 1 R0/0: rif_mgr: Gateway reachable
from Active
*Aug 29 20:45:08.545: %IOSXE_REDUNDANCY-6-PEER: Active detected chassis 2 as standby.
*Aug 29 20:45:08.540: %STACKMGR-6-STANDBY_ELECTED: Chassis 1 R0/0: stack_mgr: Chassis 2 has been
elected STANDBY.
```

Figure 10-9 Standby election

```
*Aug 29 20:44:07.123: %EWLC_HA_LIB_MESSAGE-6-BULK_SYNC_STATE_INFO: Chassis 1 R0/0: wncmgrd: INFO: Bulk
sync status : COLD
```

Figure 10-10 Standby comes up cold prior to the bulk sync

```
*Aug 29 20:45:17.087: %VOICE_HA-7-STATUS: NONE->SSO; SSO mode will not take effect until after a
platform reload.
*Aug 29 20:45:18.589: %REDUNDANCY-5-PEER_MONITOR_EVENT: Active detected a standby insertion
(raw-event=PEER_FOUND(4))

*Aug 29 20:45:18.589: %REDUNDANCY-5-PEER_MONITOR_EVENT: Active detected a standby insertion
(raw-event=PEER_REDUNDANCY_STATE_CHANGE(5))

*Aug 29 20:45:20.149: Syncing vlan database
*Aug 29 20:45:20.165: Vlan Database sync done from bootflash:vlan.dat to stby-bootflash:vlan.dat (616
bytes)
*Aug 29 20:45:30.534: %CRYPTO_ENGINE-4-CSDL_COMPLIANCE_RSA_WEAK_KEYS: RSA keypair
CISCO_IDEVID_SUDI_LEGACY is in violation of Cisco security compliance guidelines and will be rejected
by future releases.
*Aug 29 20:45:51.774: %SMART_LIC-3-COMM_FAILED: Communications failure with the Cisco Smart License
Utility (CSLU) : Unable to resolve server hostname/domain name
*Aug 29 20:47:21.226: %RIF_MGR_FSM-6-RMI_LINK_UP: Chassis 1 R0/0: rif_mgr: The RMI link is UP.
*Aug 29 20:47:23.646: %HA_CONFIG_SYNC-6-BULK_CFGSYNC_SUCCEED: Bulk Sync succeeded
*Aug 29 20:47:23.686: %VOICE_HA-7-STATUS: VOICE HA bulk sync done.
*Aug 29 20:47:24.725: %RF-5-RF_TERMINAL_STATE: Terminal state reached for (SSO)
```

Figure 10-11 Standby comes up hot after the bulk sync

```
9800L#redundancy force-switchover
Proceed with switchover to standby RP? [confirm]
    Manual Swact = enabled

Chassis 1 reloading, reason - Non participant detected

*Aug 31 07:26:52.680: RMI-GW-NOTICE: Forced switchover notification received

*Aug 31 07:26:53.782: %RF-5-RF_RELOAD: Self reload. Reason: redundancy force-switchover
Aug 31 07:26:54.805: %PMAN-5-EXITACTION: C0/0: pvp: Process manager is exiting:
Aug 31 07:26:54.842: %PMAN-5-EXITACTION: F0/0: pvp: Process manager is exiting:

*Aug 31 07:26:54.060: %SYS-5-SWITCHOVER: Switchover requested by red_switchover_process. Reason:
redundancy force-switchover.Aug 31 07:27:00.884: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is
exiting: process exit with reload fru code
```

Figure 10-12 SSO switchover on the active C9800

```
9800L-stby#Ewlc: triggered dual-active recovery,setting hostname to 9800L, Mode: 4
*Aug 31 07:26:53.881: %REDUNDANCY-3-SWITCHOVER: RP switchover (PEER_NOT_PRESENT)
*Aug 31 07:26:53.881: %REDUNDANCY-3-REDUNDANCY_ALARMS: Unable to assert REDUNDANCY alarm

*Aug 31 07:26:53.881: %REDUNDANCY-3-REDUNDANCY_ALARMS: Unable to assert REDUNDANCY alarm

*Aug 31 07:26:53.881: %REDUNDANCY-3-SWITCHOVER: RP switchover (PEER_DOWN)
*Aug 31 07:26:53.881: %REDUNDANCY-3-SWITCHOVER: RP switchover (PEER_REDUNDANCY_STATE_CHANGE)
*Aug 31 07:26:54.164: SNMPHA-CHKPT: chkpt: msg is NULL

*Aug 31 07:26:54.556: SNMPHA-CHKPT: chkpt: msg is NULL

*Aug 31 07:26:54.676: WLC-HA-Notice: RF Progression event: RF_PROG_ACTIVE_FAST, Switchover triggered
*Aug 31 07:26:54.681: %LINK-3-UPDOWN: Interface Lsmpi0, changed state to up
*Aug 31 07:26:54.681: %LINK-3-UPDOWN: Interface EOBC0, changed state to up
*Aug 31 07:26:54.710: RMI-HAINFRA-INFO: Configured primary IP 192.168.1.5/255.255.255.0 on active(mgmt)
*Aug 31 07:26:54.710: RMI-HAINFRA-INFO: Configured secondary IP 192.168.1.17/255.255.255.0 on
active(mgmt)
*Aug 31 07:26:54.731: %VOICE_HA-2-SWITCHOVER_IND: SWITCHOVER, from STANDBY_HOT to ACTIVE state.
```

Figure 10-13 SSO switchover on the standby C9800

```
9800L#show chassis rmi
Chassis/Stack Mac Address : d478.9b3c.5e80 - 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                RMI-IP
--------------------------------------------------------------------------------------------------------
*1       Active   d478.9b3c.5e80     1      V02     Ready                169.254.1.15       192.168.1.15
 2       Standby  d478.9b3c.5f60     1      V02     Ready                169.254.1.17       192.168.1.17


9800L#show redundancy states
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = sso
     Maintenance Mode = Disabled
    Manual Swact = enabled
 Communications = Up

   client count = 150
 client_notification_TMR = 30000 milliseconds
           RF debug mask = 0x0
Gateway Monitoring = Enabled
Gateway monitoring interval  = 8 secs
```

Figure 10-14 HA monitoring from the active C9800

```
9800L-stby#show chassis rmi
Chassis/Stack Mac Address : d478.9b3c.5e80 - 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                RMI-IP
--------------------------------------------------------------------------------------------------------
 1       Active   d478.9b3c.5e80     1      V02     Ready                169.254.1.15       192.168.1.15
*2       Standby  d478.9b3c.5f60     1      V02     Ready                169.254.1.17       192.168.1.17



9800L-stby#show redundancy states
       my state = 8  -STANDBY HOT
     peer state = 13 -ACTIVE
           Mode = Duplex
           Unit = Primary
        Unit ID = 2

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = sso
     Maintenance Mode = Disabled
    Manual Swact = cannot be initiated from this the standby unit
 Communications = Up

   client count = 150
 client_notification_TMR = 30000 milliseconds
           RF debug mask = 0x0
Gateway Monitoring = Enabled
Gateway monitoring interval  = 8 secs
```

Figure 10-15 HA monitoring from the standby C9800

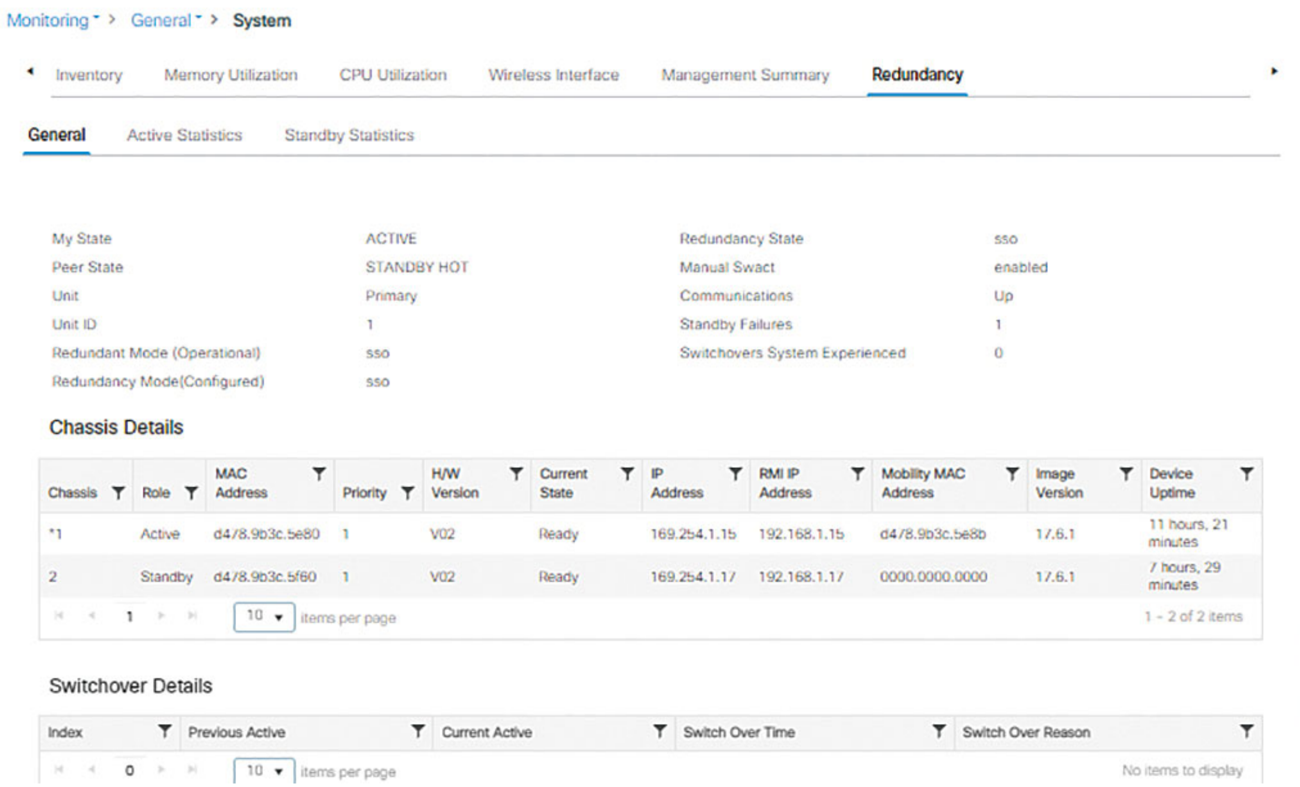


Figure 10-16 HA state monitoring from the active C9800 GUI

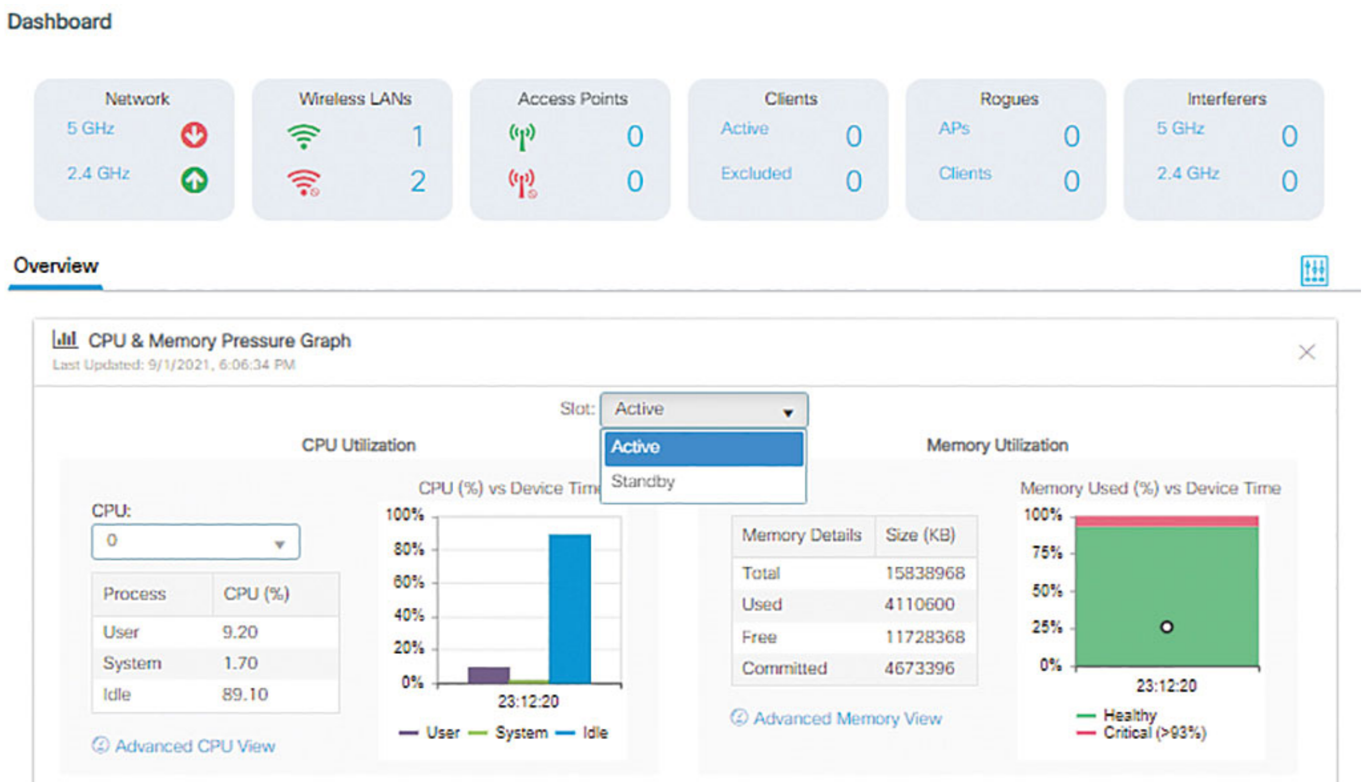


Figure 10-17 HA active and standby resource monitoring from the active C9800 GUI

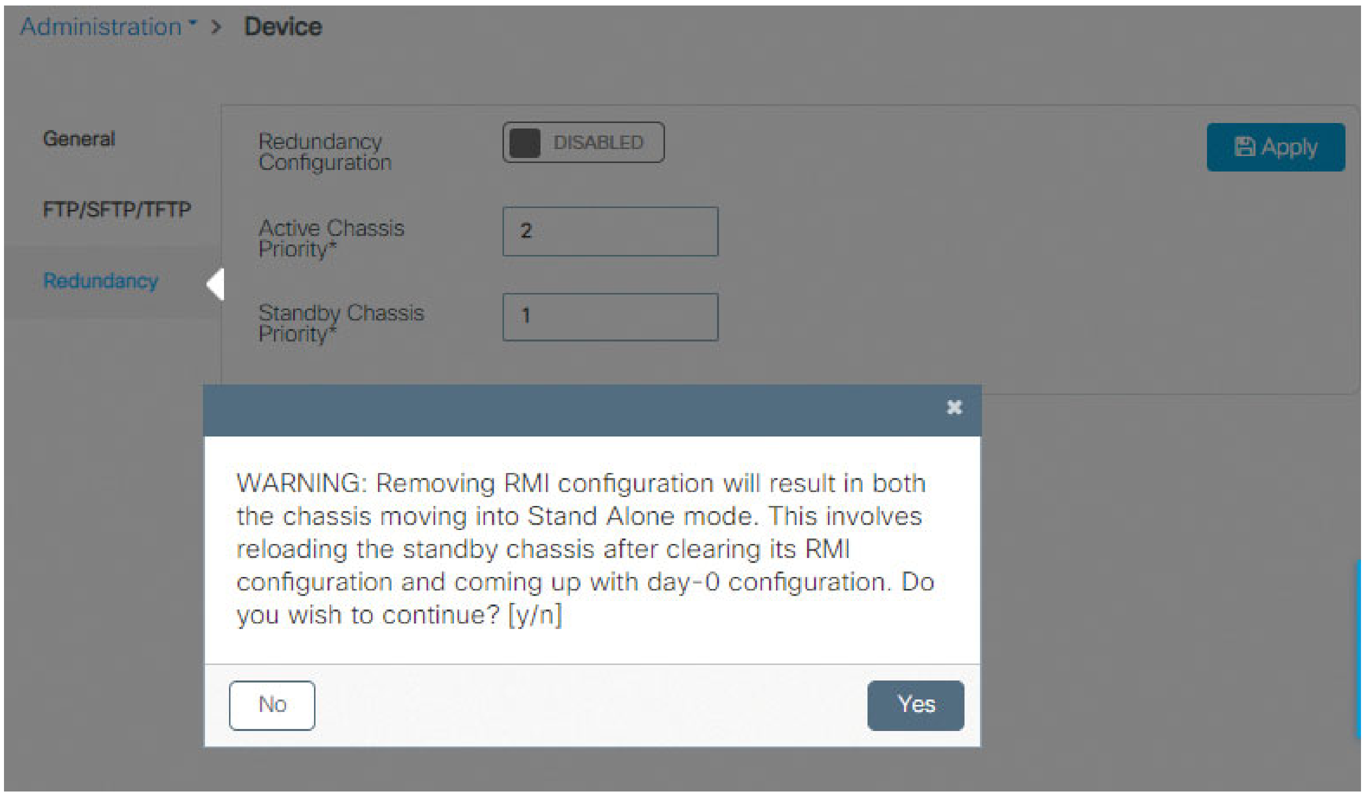


Figure 10-18 Disabling HA SSO

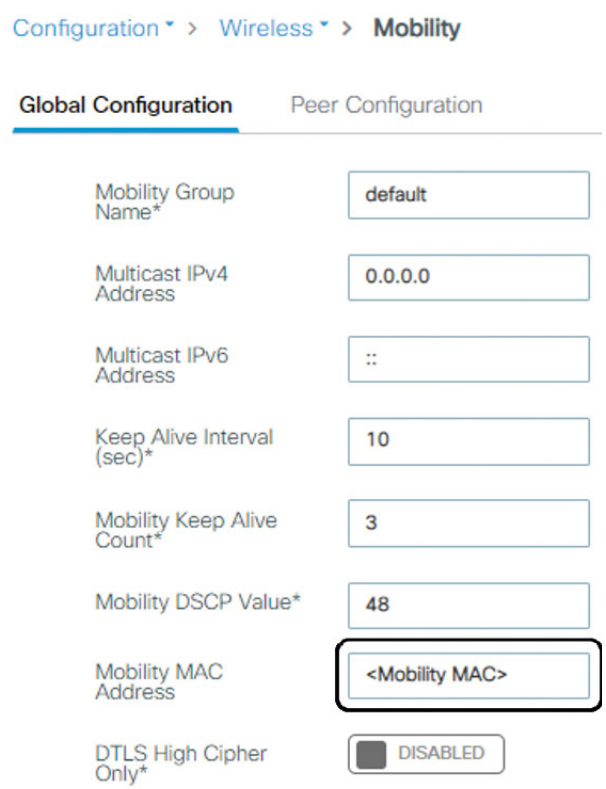


Figure 10-19 Mobility MAC Configuration

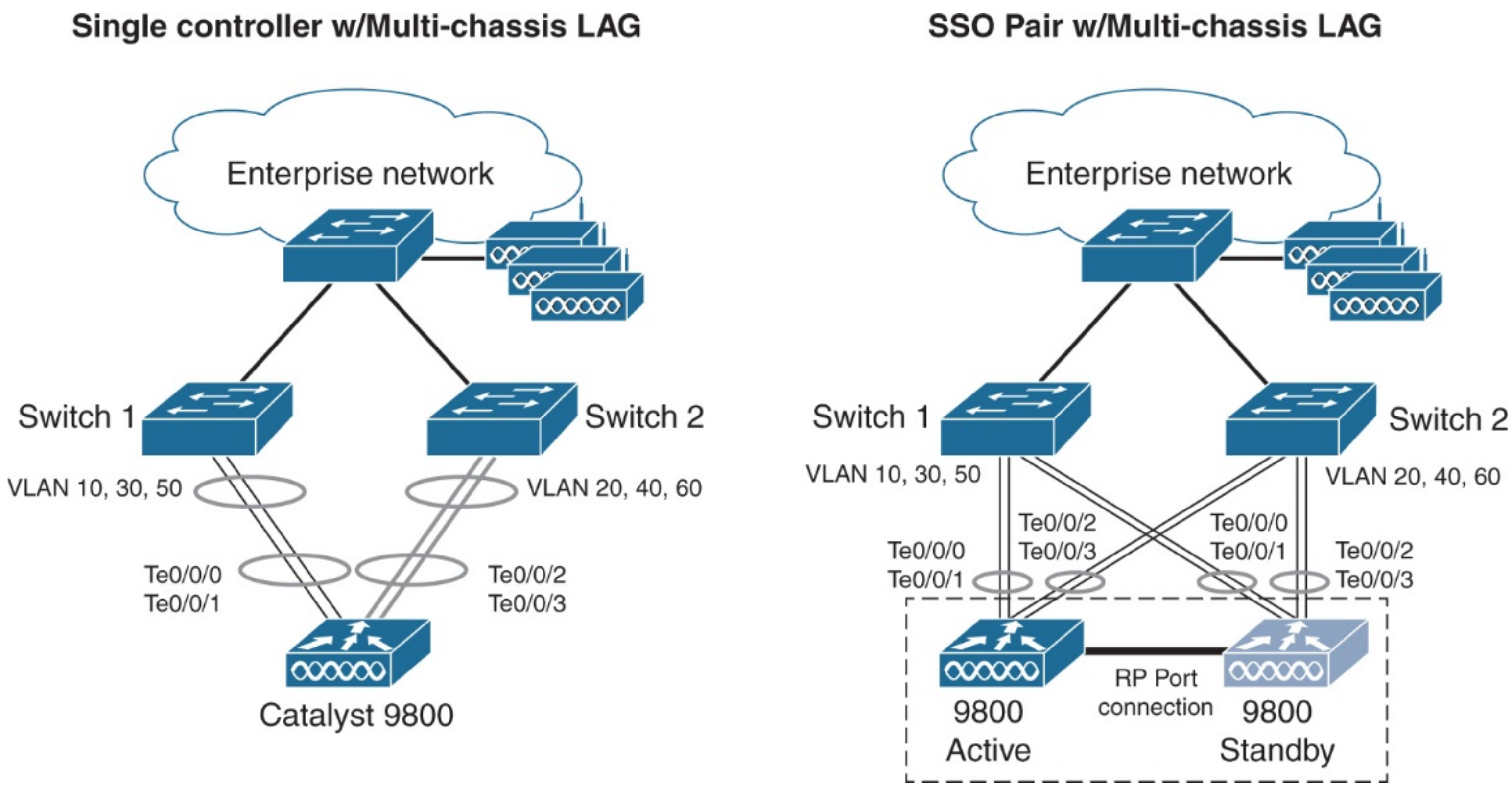


Figure 10-20 C9800 to switch connections with Multi-Chassis LAG

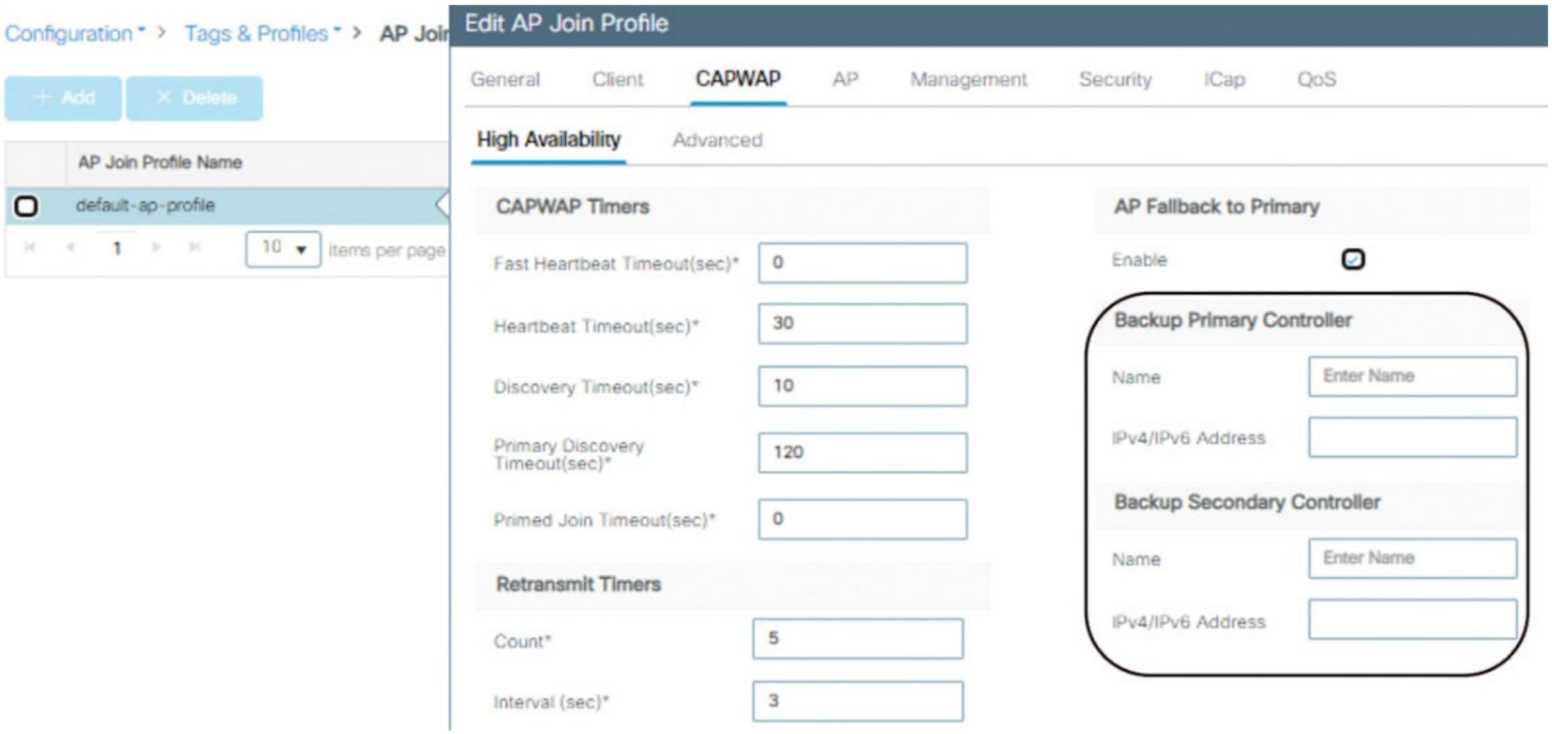


Figure 10-21 N+1 HA configuration on access points

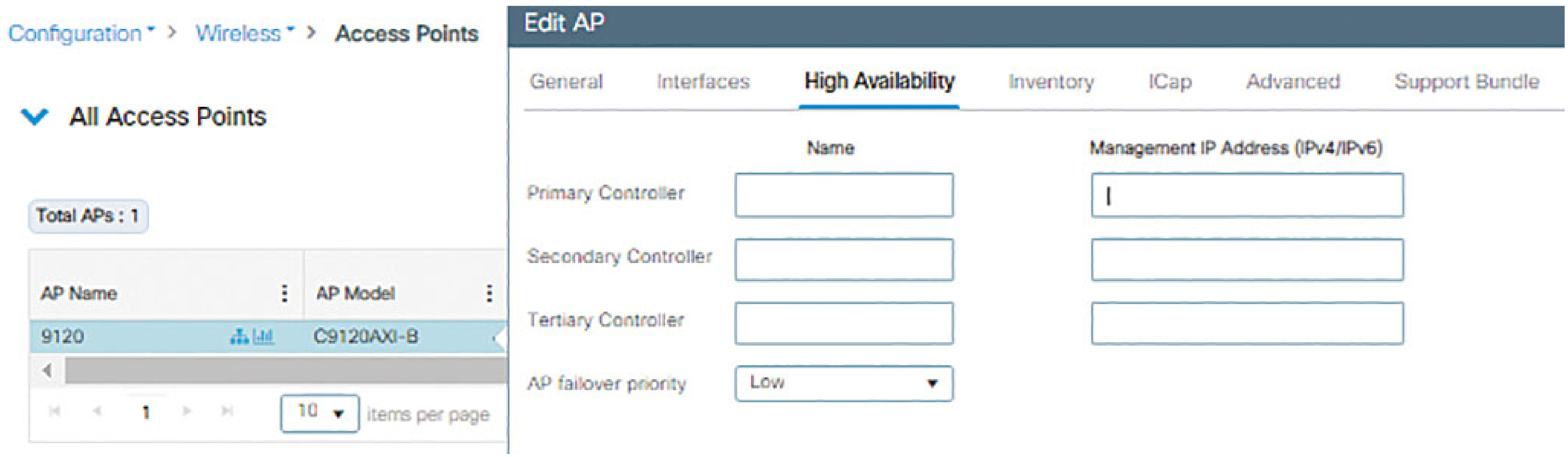


Figure 10-22 N+1 HA Configuration on AP Join Profile

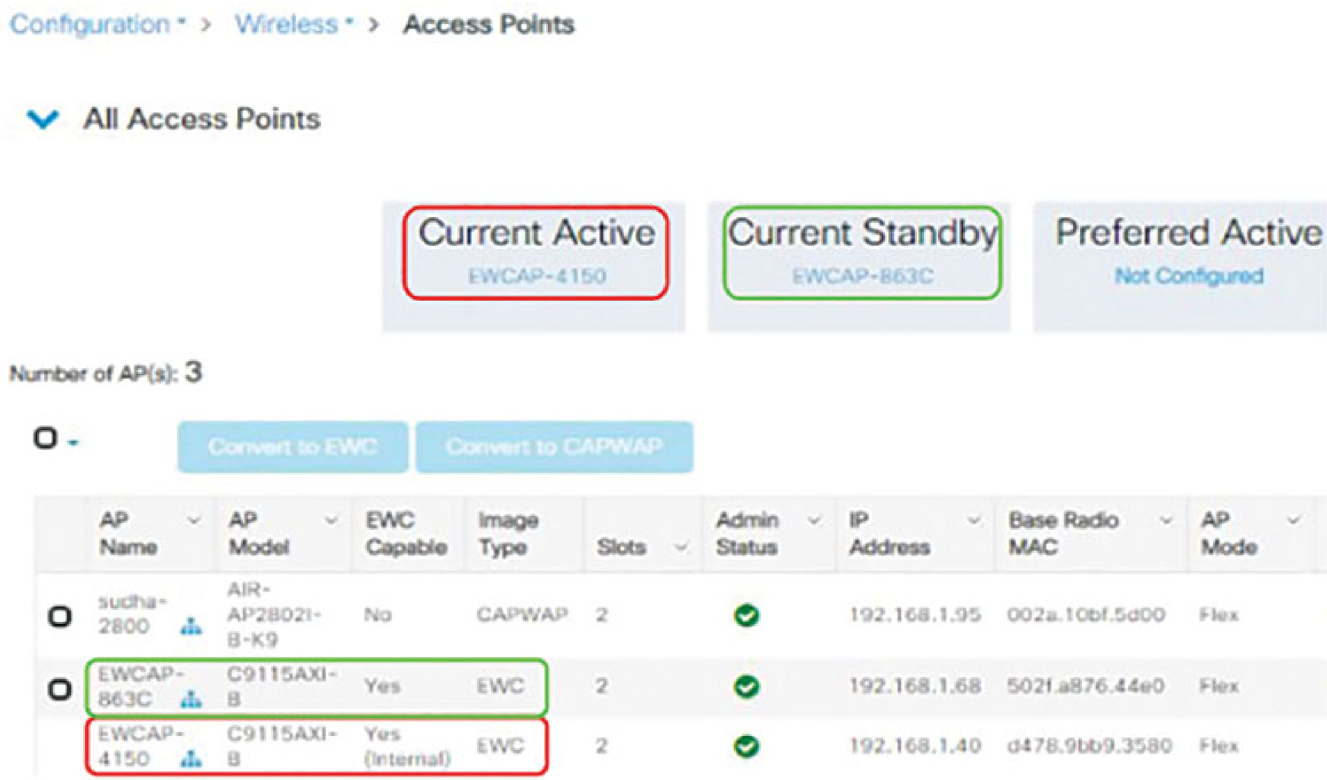


Figure 10-23 Active and Standby EWC-APs

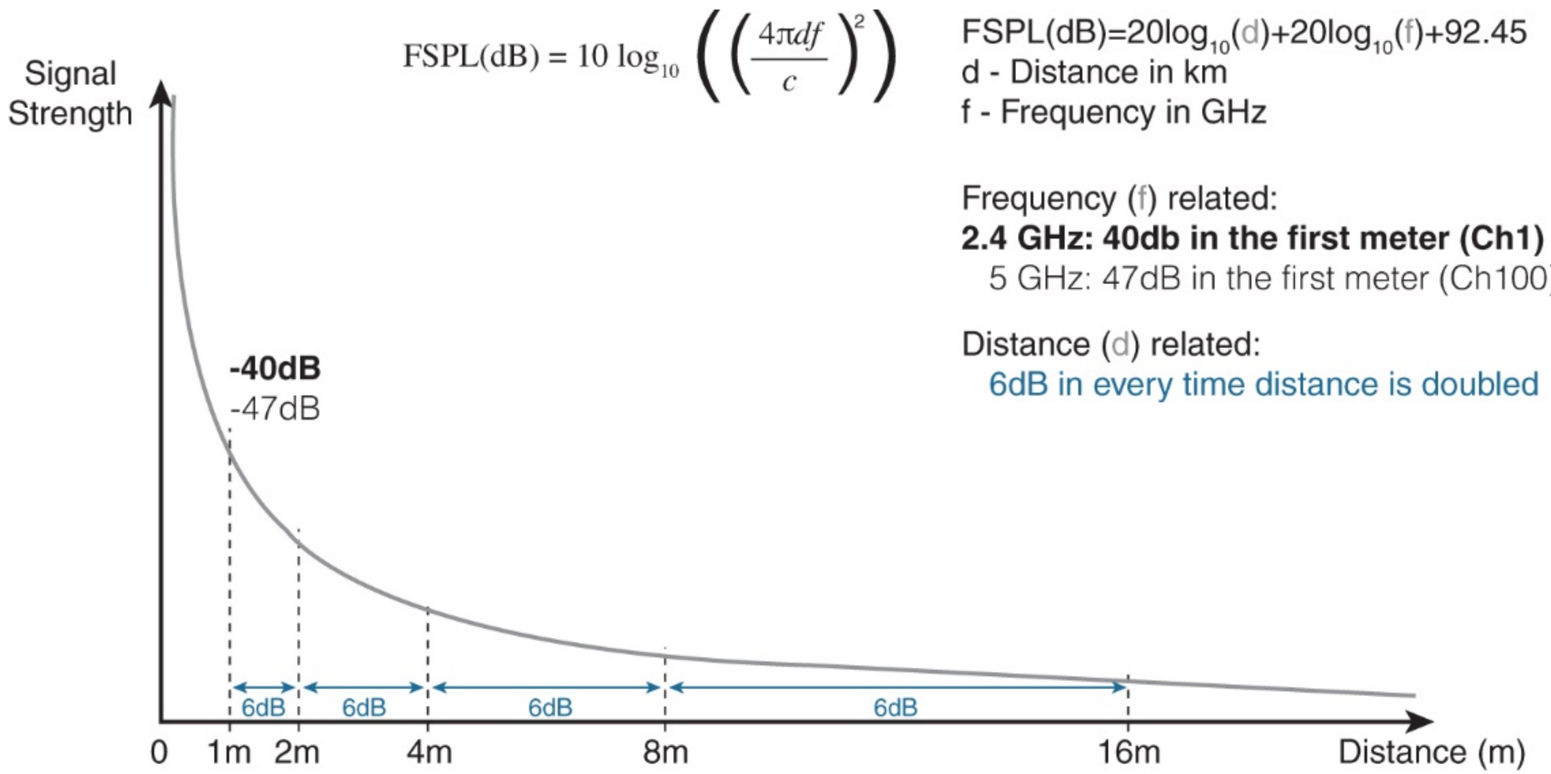


Figure 11-1 Free space path loss; a 6 dB decrease every time distance is doubled

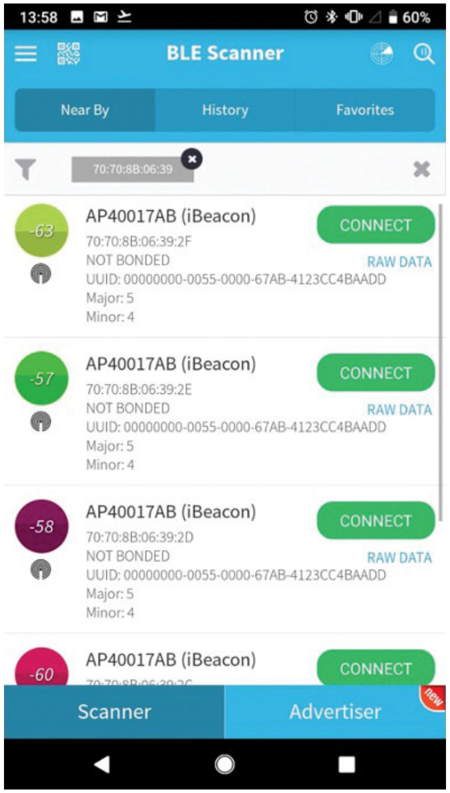


Figure 11-2 A BLE scanner application running on a smartphone

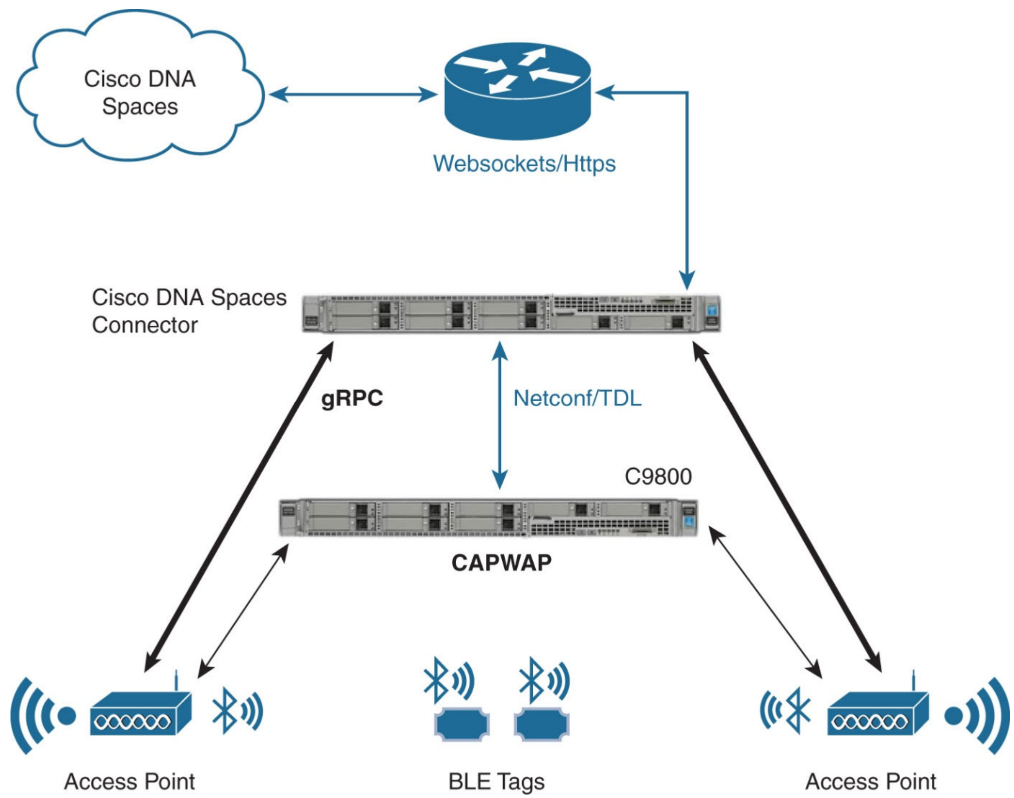


Figure 11-3 Cisco DNA Spaces Bluetooth telemetry architecture

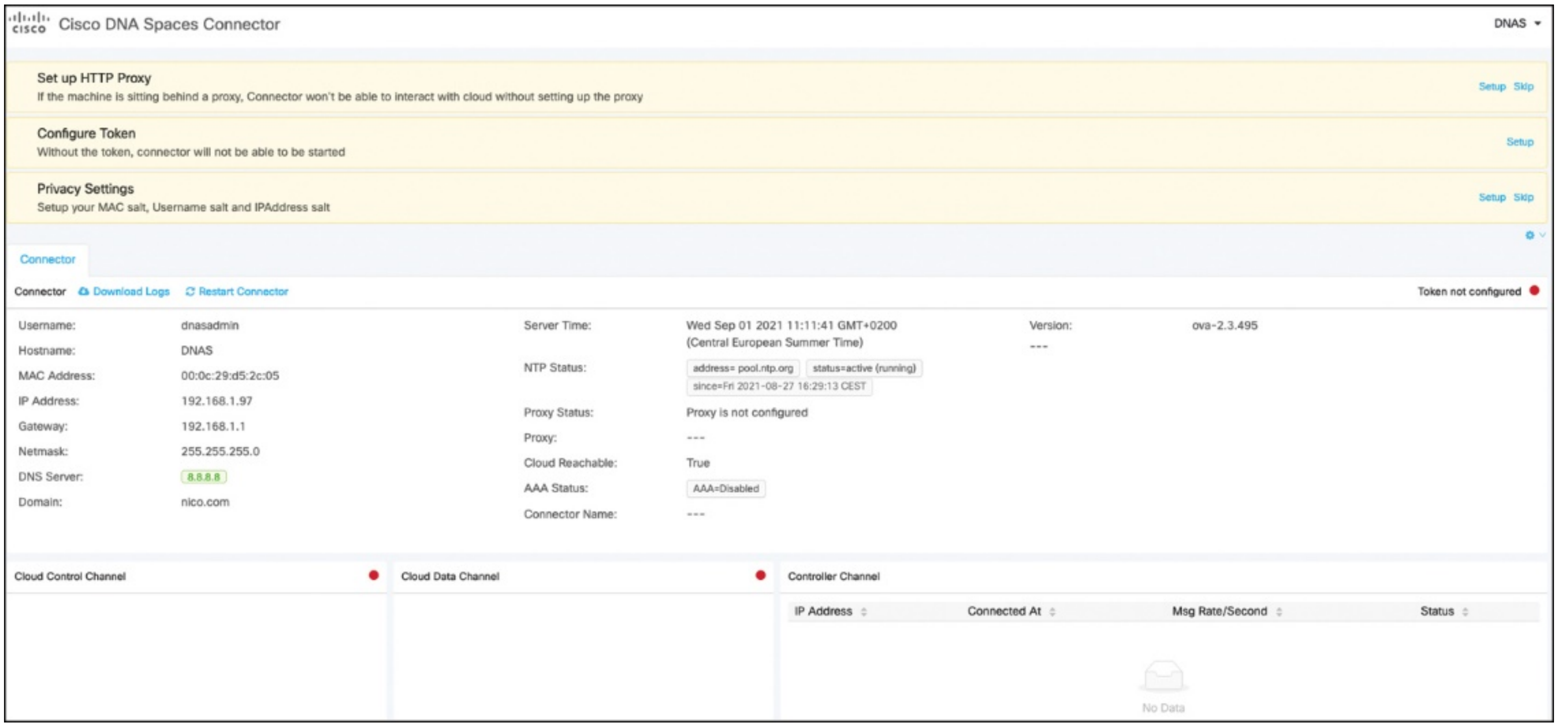


Figure 11-4 CiscoDNA Spaces connector initial WebUI screen

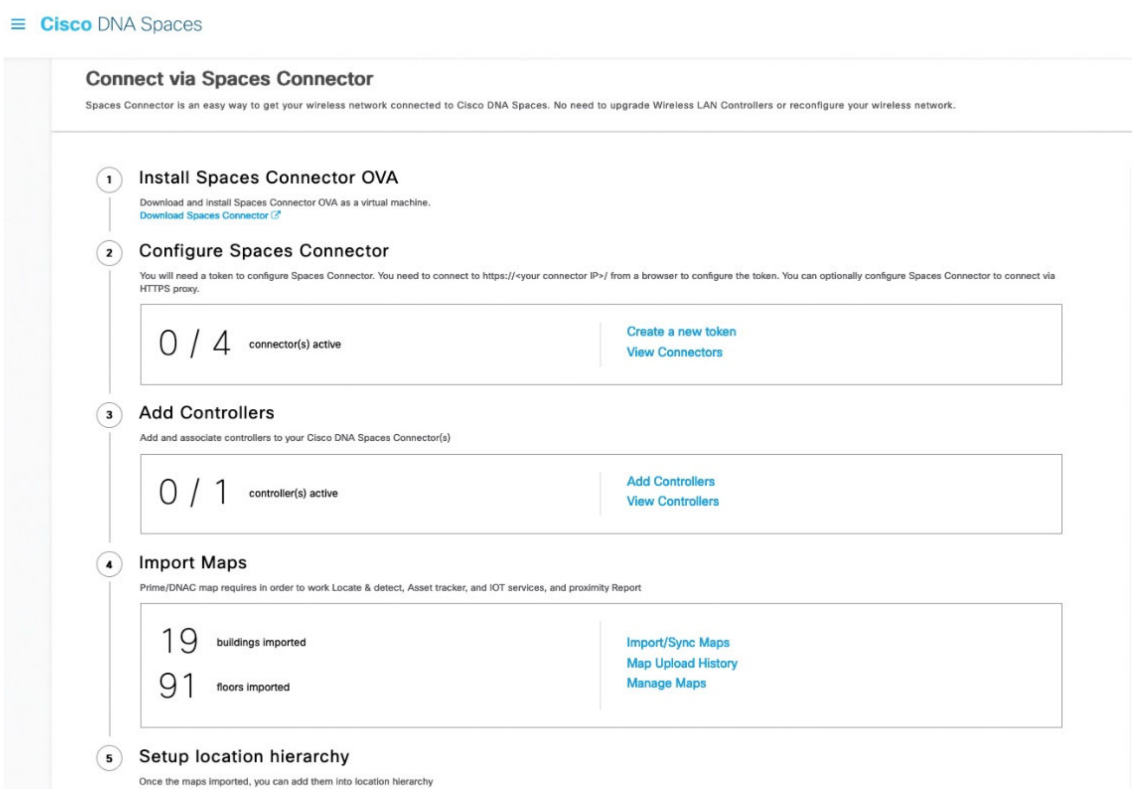


Figure 11-5 DNA Spaces Setup >Wireless page allowing you to generate tokens

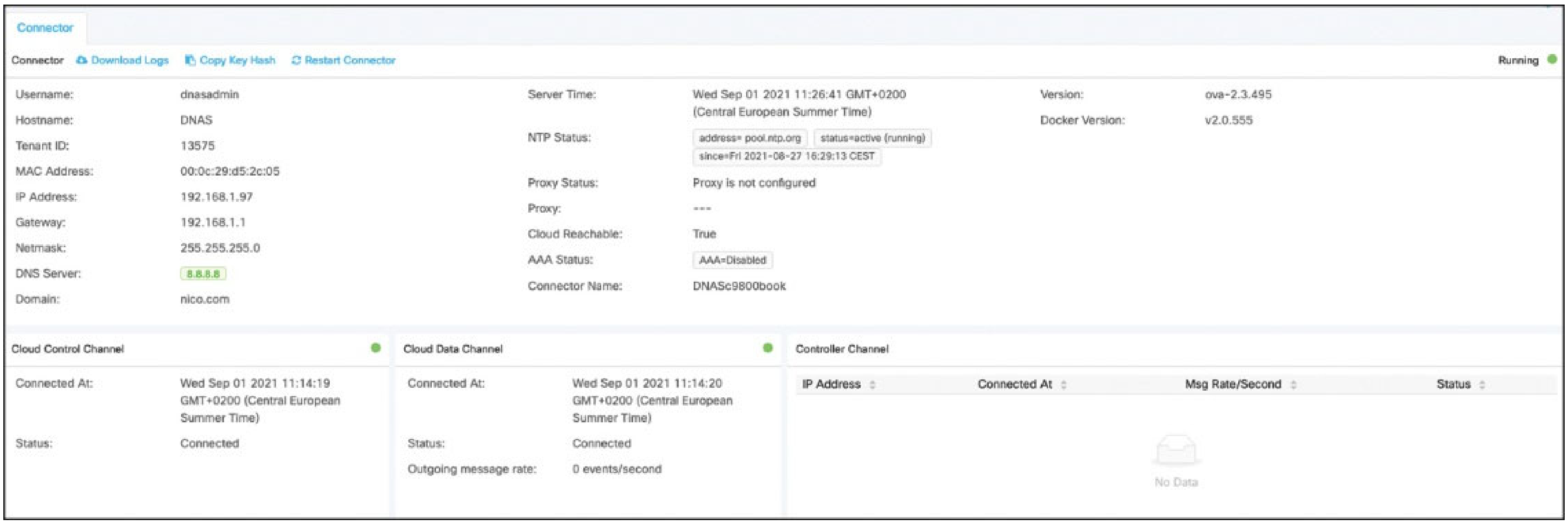


Figure 11-6 Cisco DNA Spaces WebUI page after successful connection to the cloud

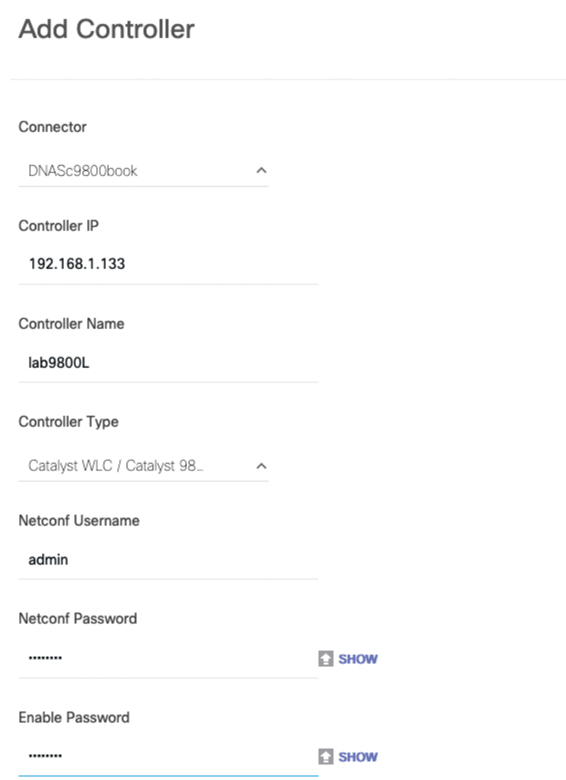


Figure 11-7 Cisco DNA Spaces connector Add Controller page

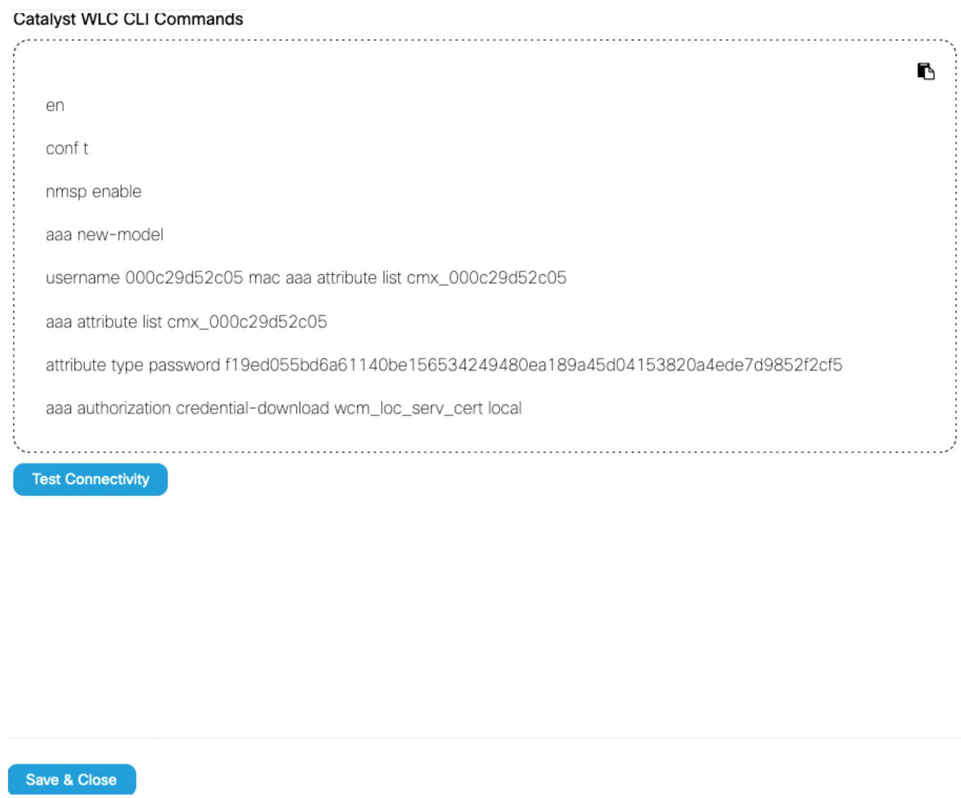


Figure 11-8 DNA Spaces connector CLI commands overview

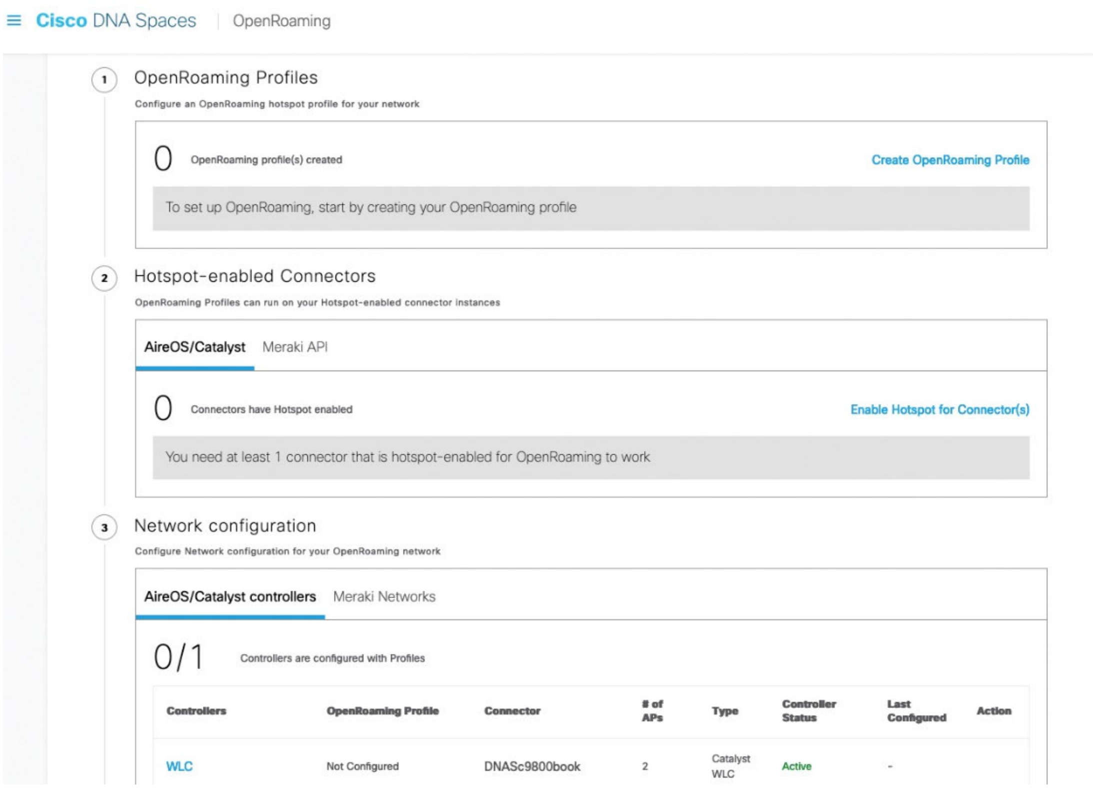


Figure 11-9 Cisco DNA Spaces OpenRoaming page

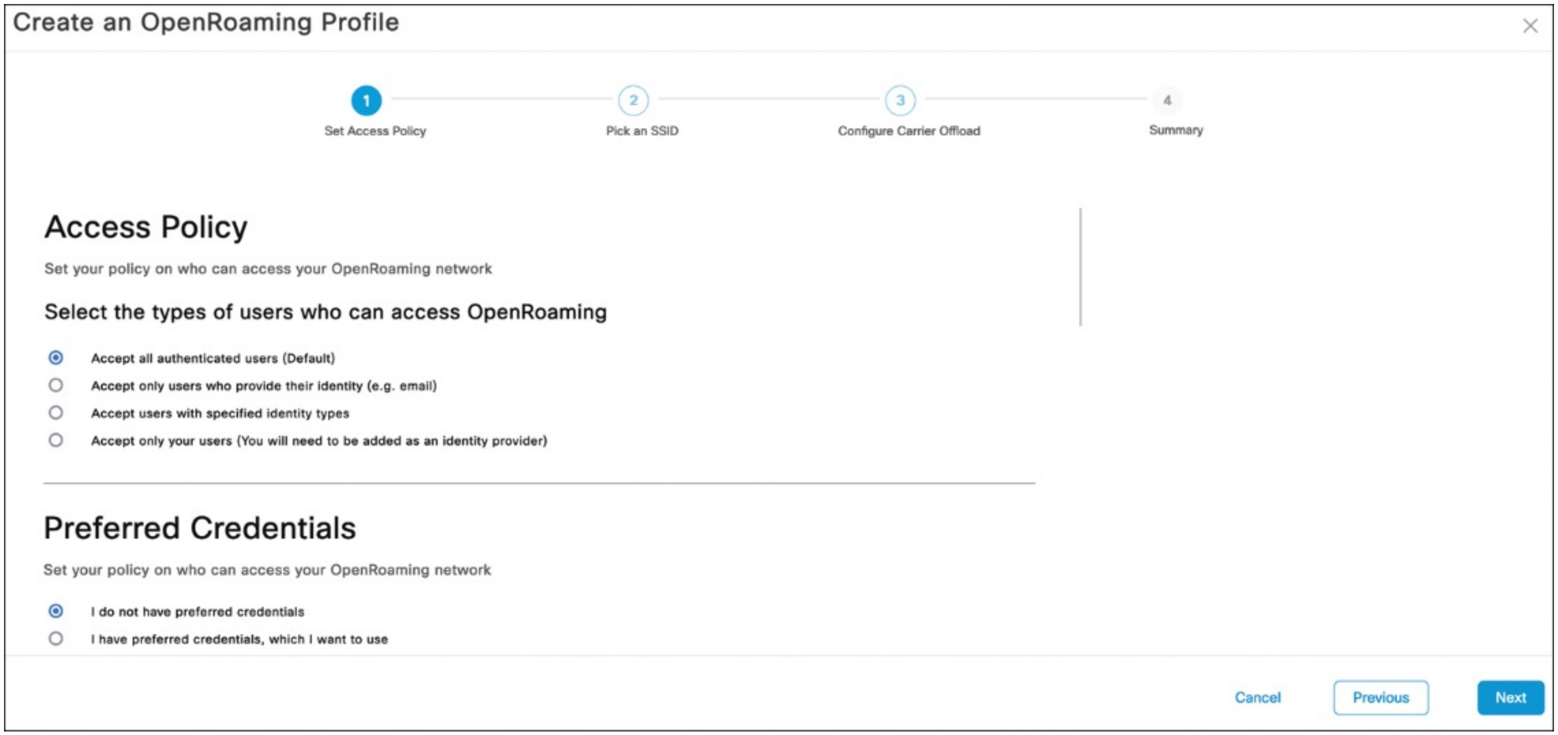


Figure 11-10 Cisco DNA Spaces OpenRoaming profile access policy page

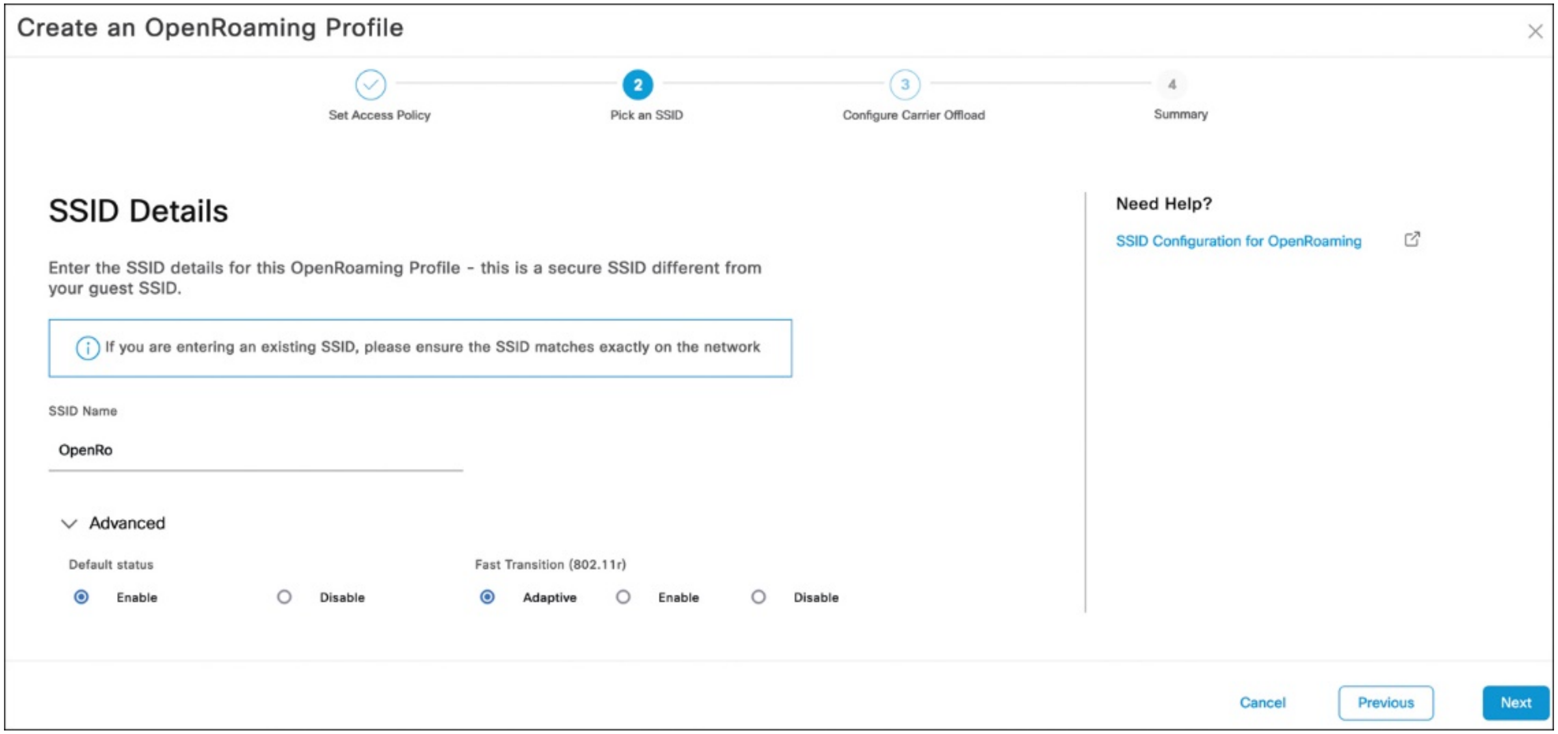


Figure 11-11 Cisco DNA Spaces OpenRoaming profile SSID configuration page

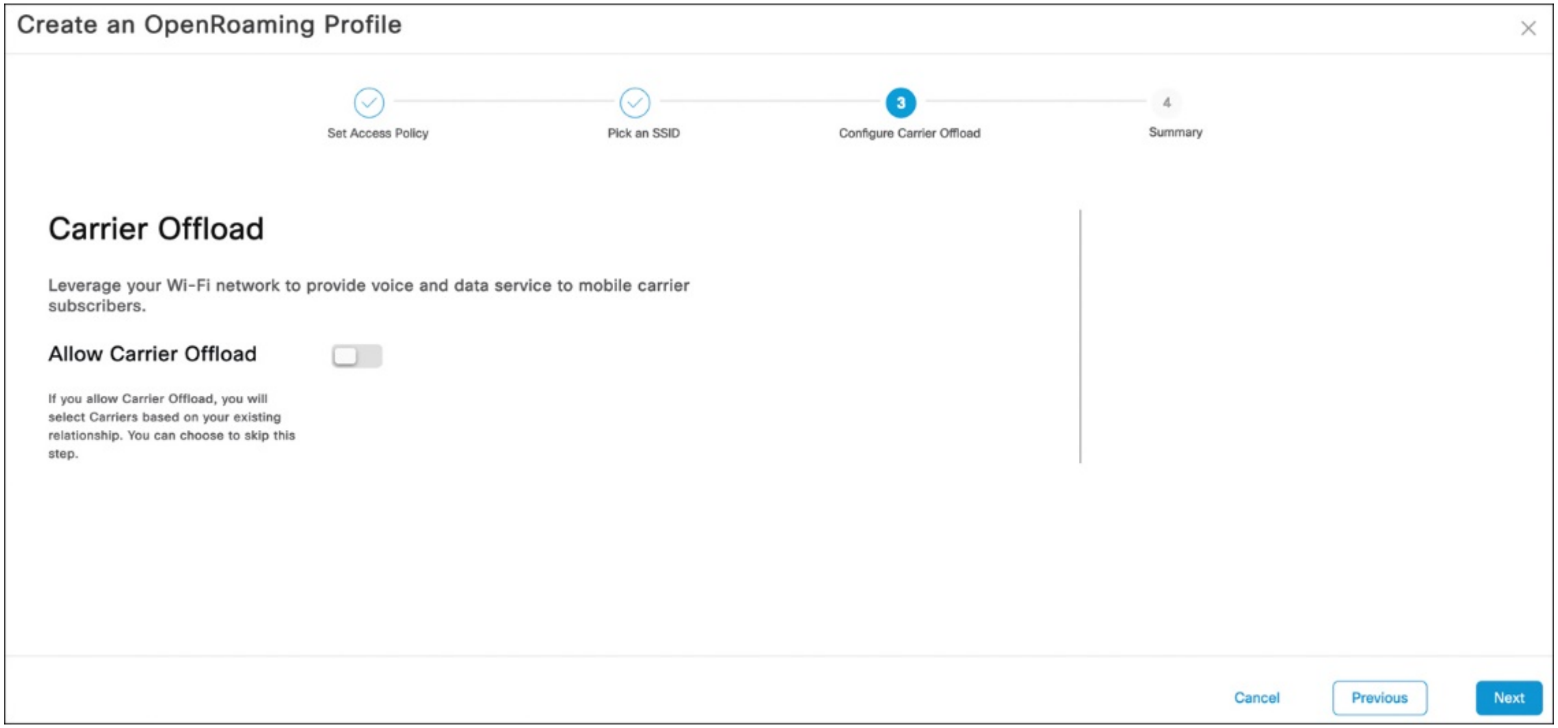


Figure 11-12 Cisco DNA Spaces OpenRoaming profile carrier offload page

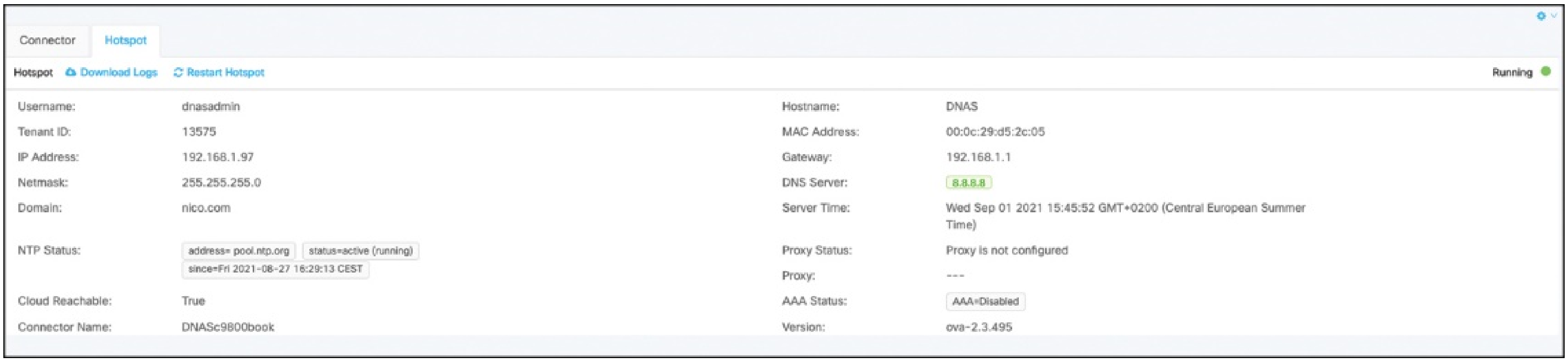


Figure 11-13 Cisco DNA Spaces home page with the hotspot tab

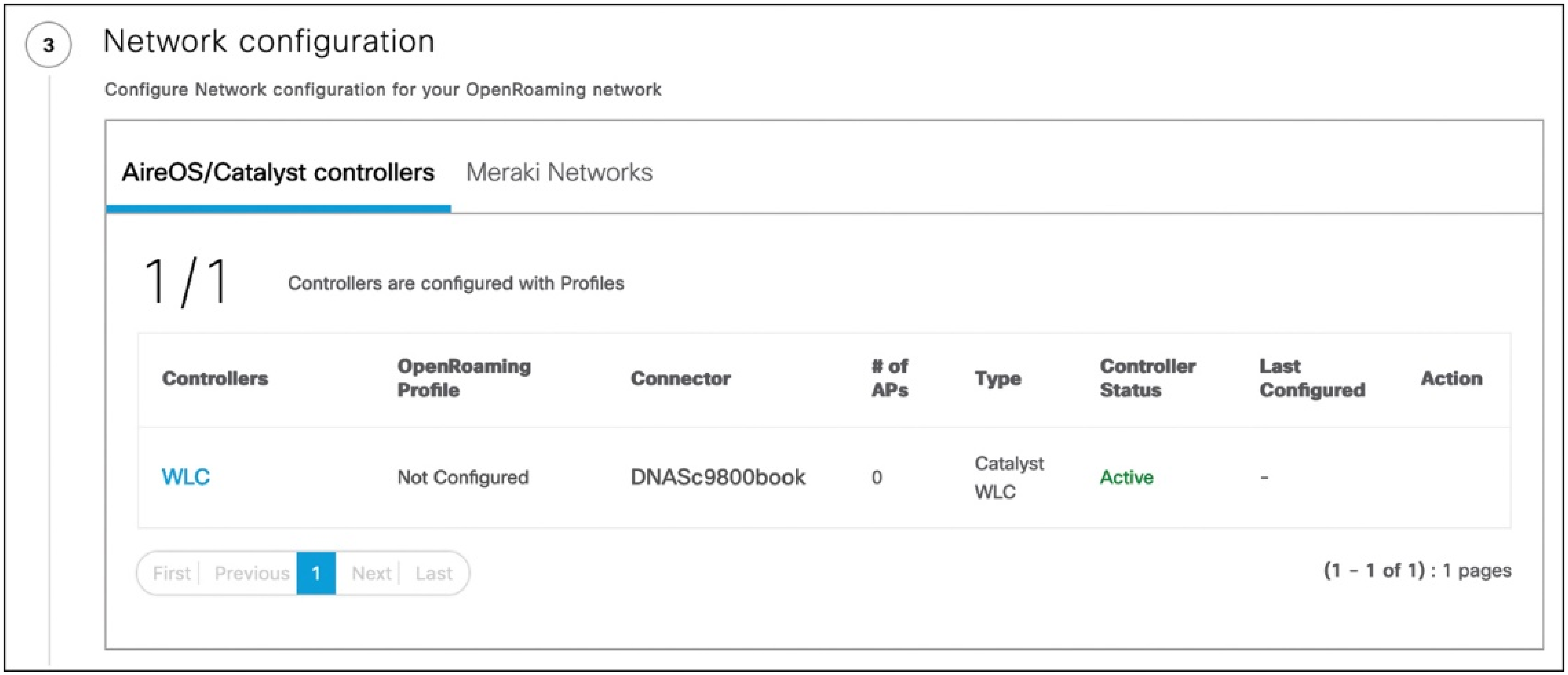


Figure 11-14 Cisco DNA Spaces network configuration section

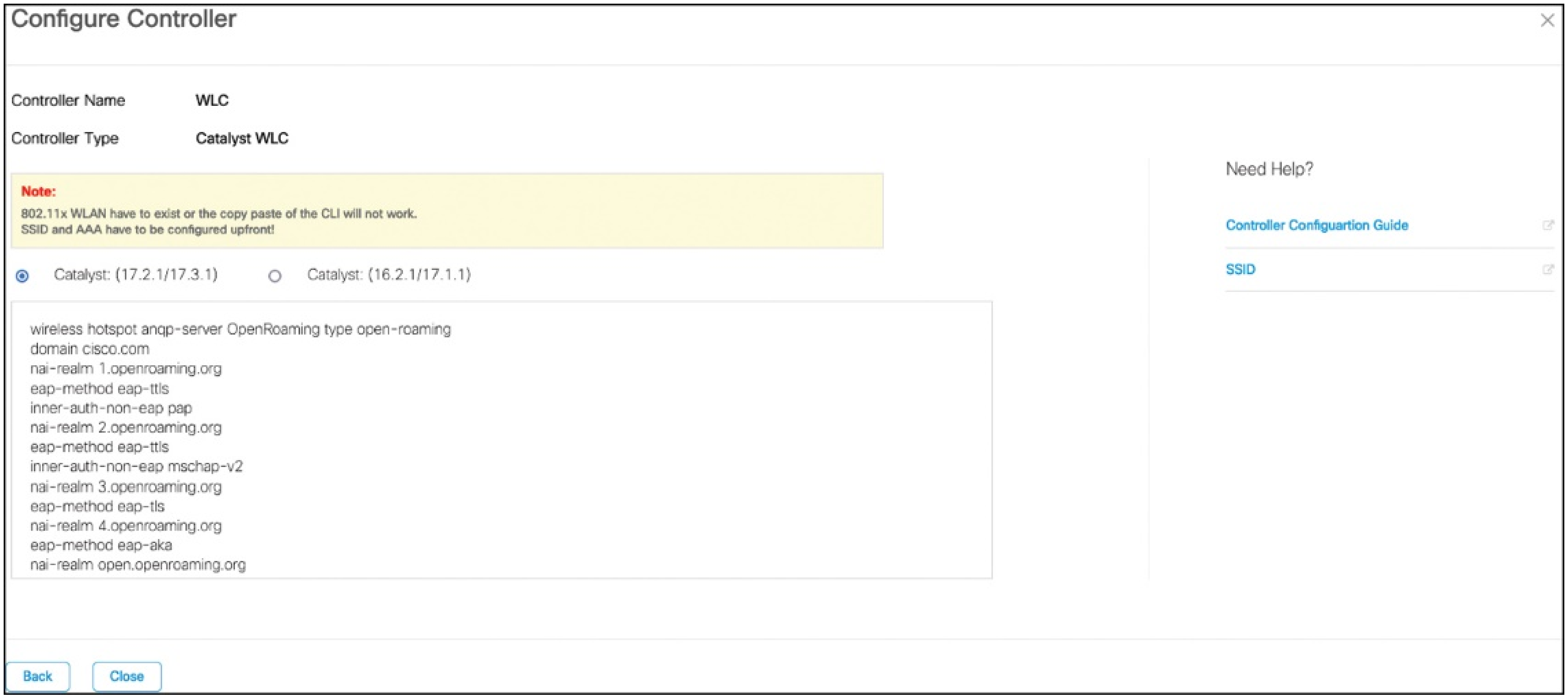


Figure 11-15 Cisco DNA Spaces controller configuration page shows you the command it pushes

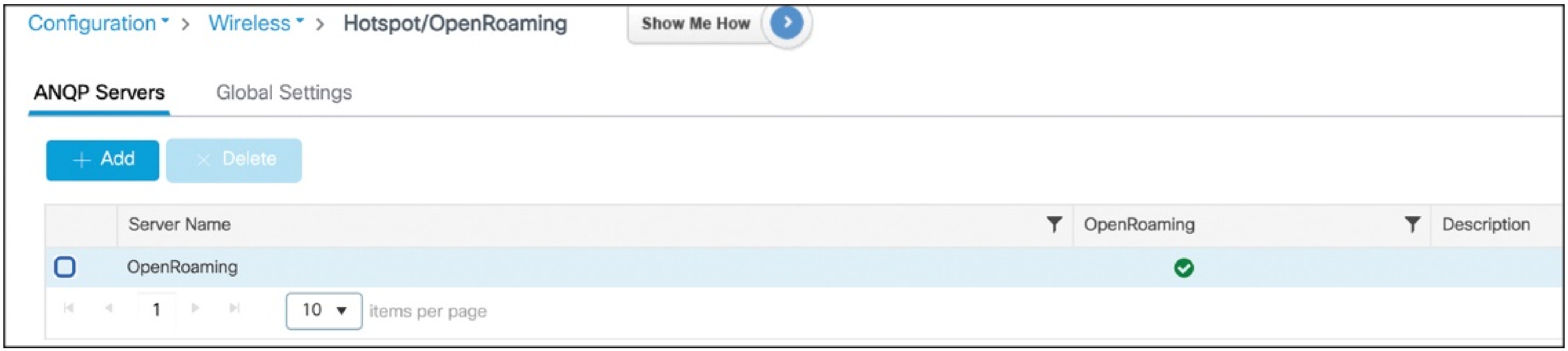


Figure 11-16 WLC OpenRoaming configuration page

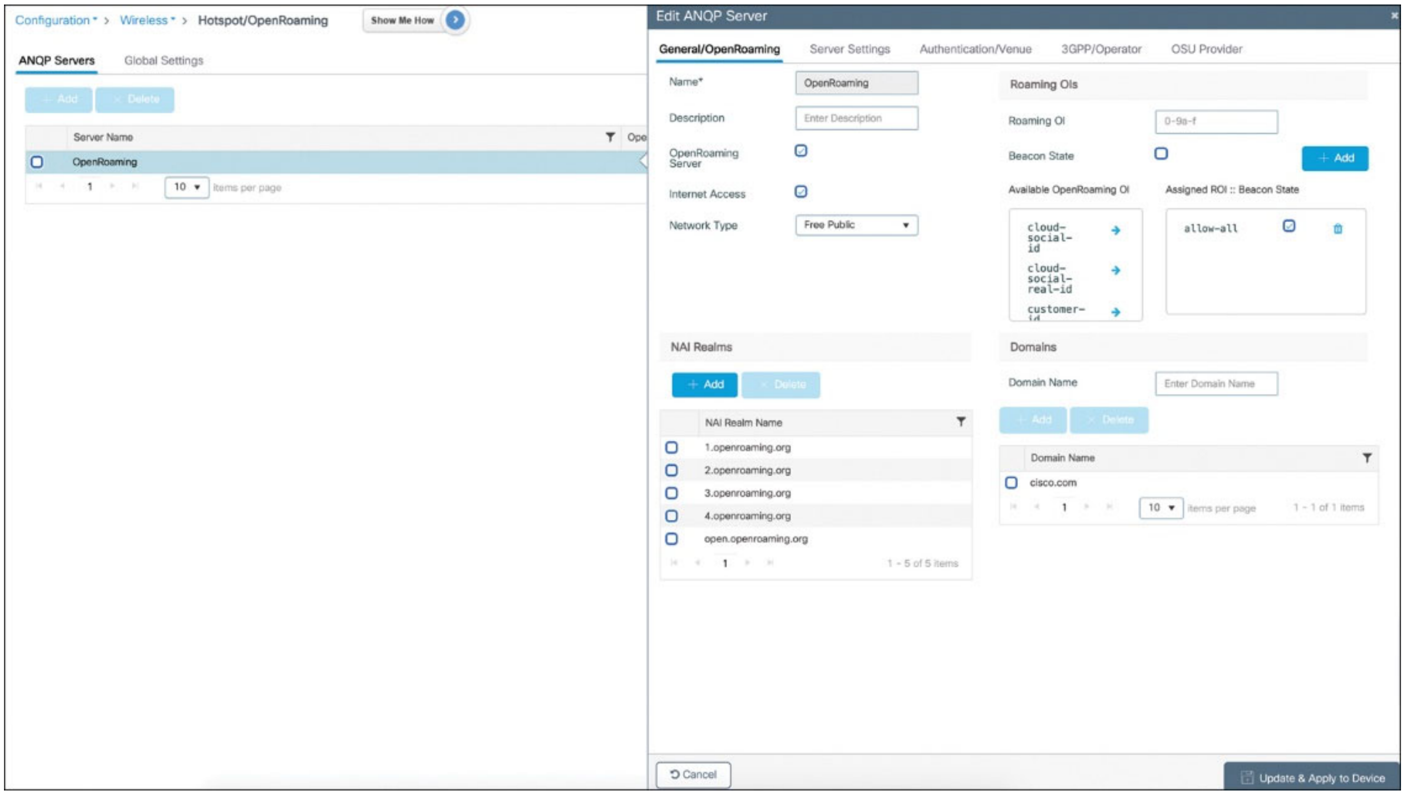


Figure 11-17 WLC OpenRoaming configuration page

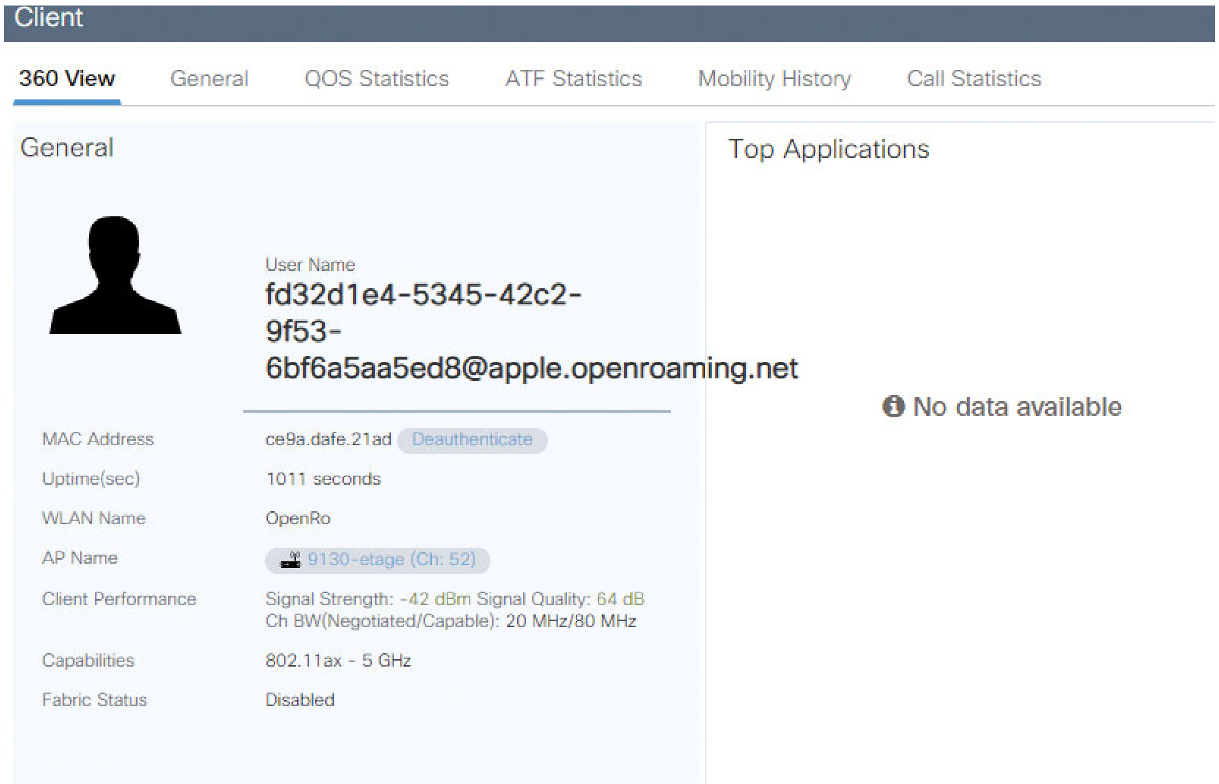


Figure 11-18 OpenRoaming user details on the 9800 WLC

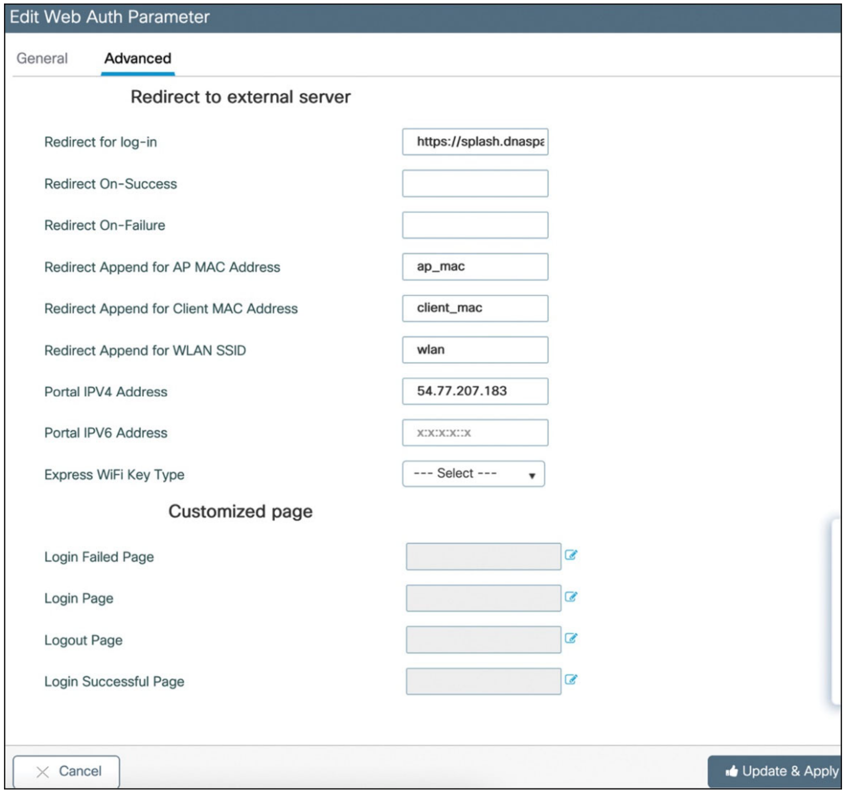


Figure 11-19 WLC advanced webauth parameter map configuration screen

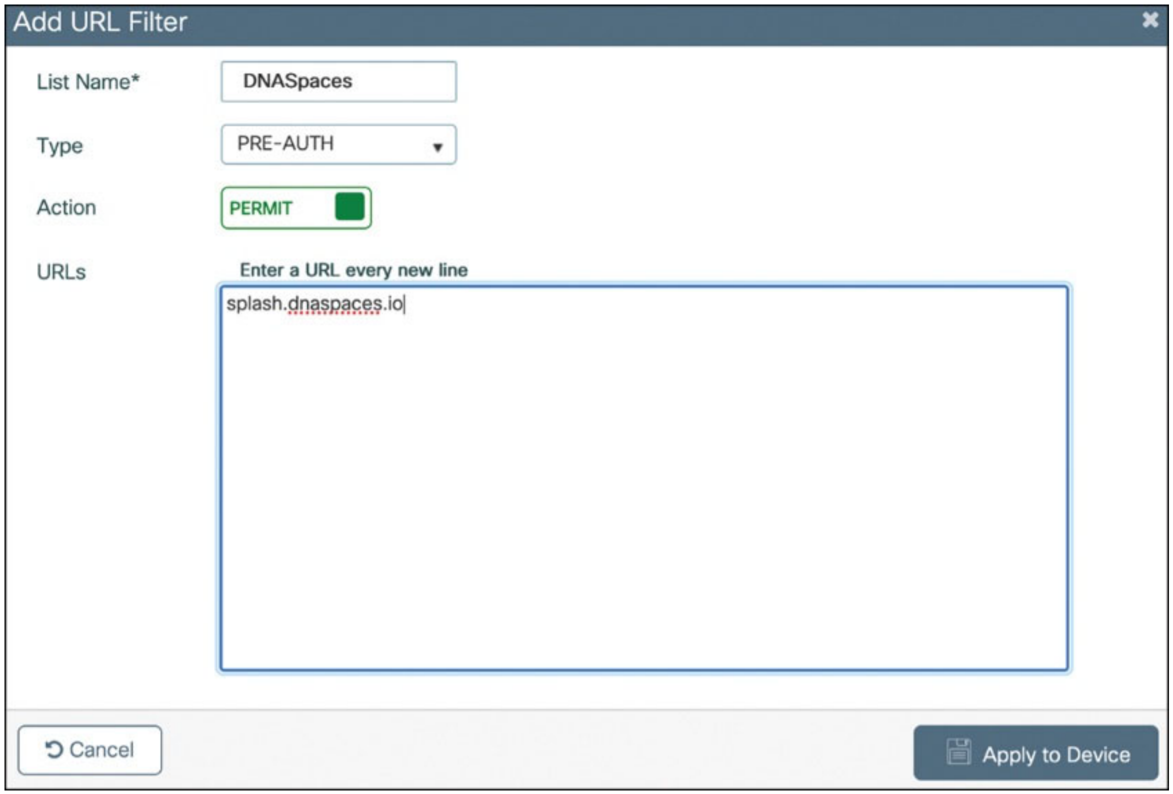


Figure 11-20 WLC URL filter allows you to define URLs accessible in the pre-auth phase

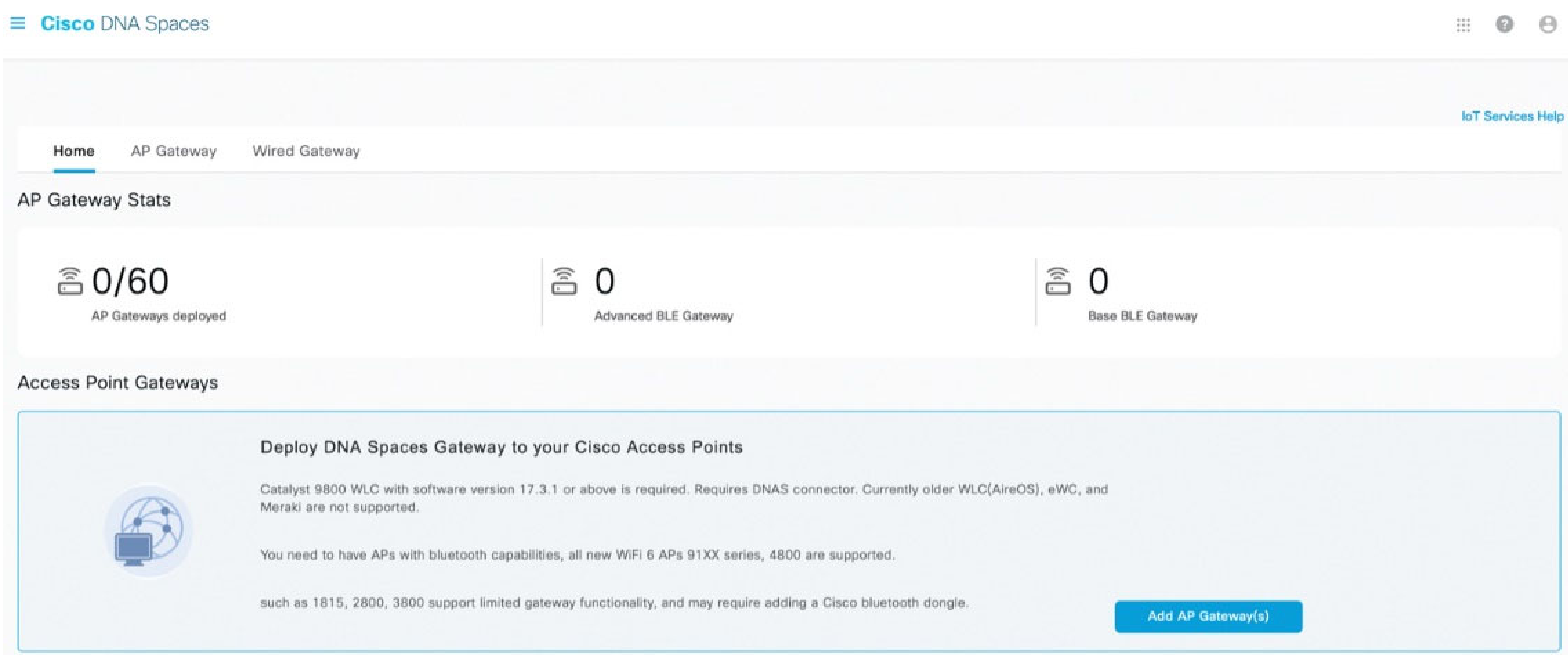


Figure 11-21 Cisco DNA Spaces IoT Services menu

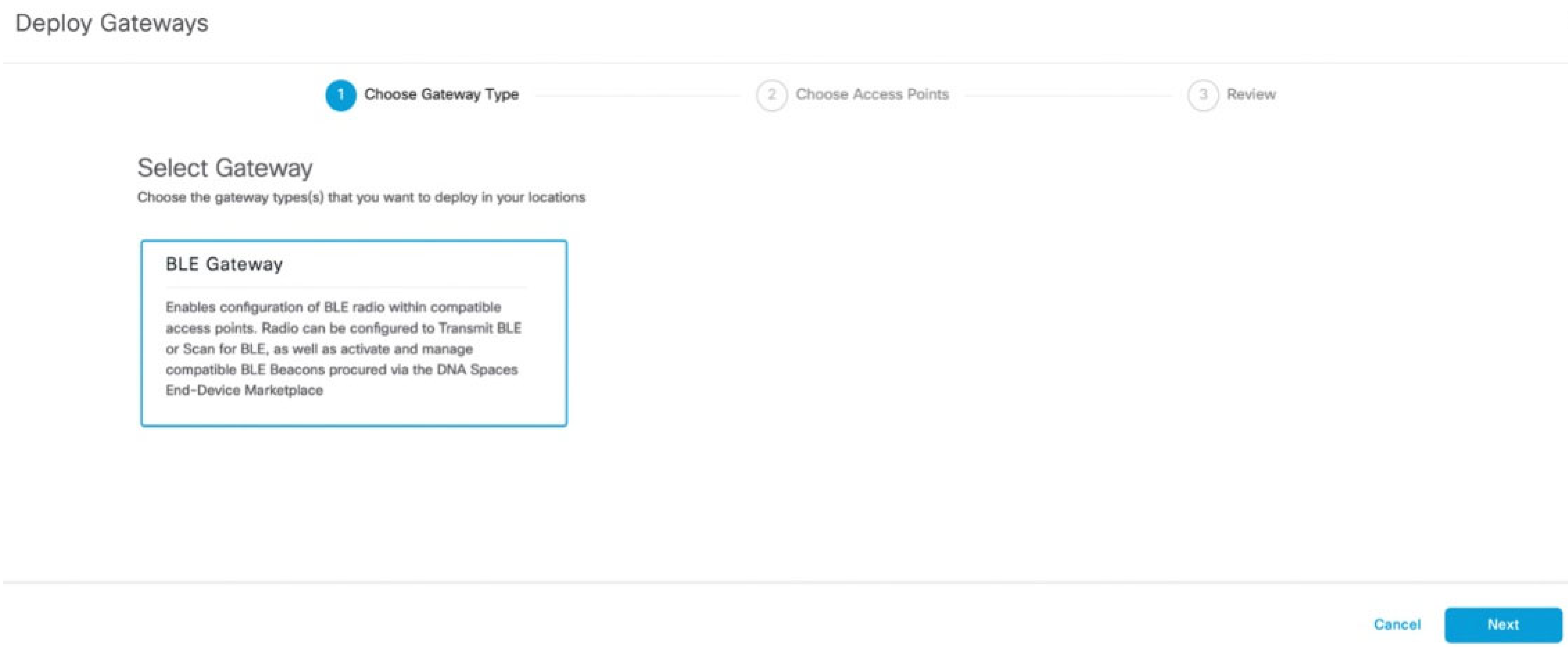


Figure 11-22 Cisco DNA Spaces BLE gateway configuration

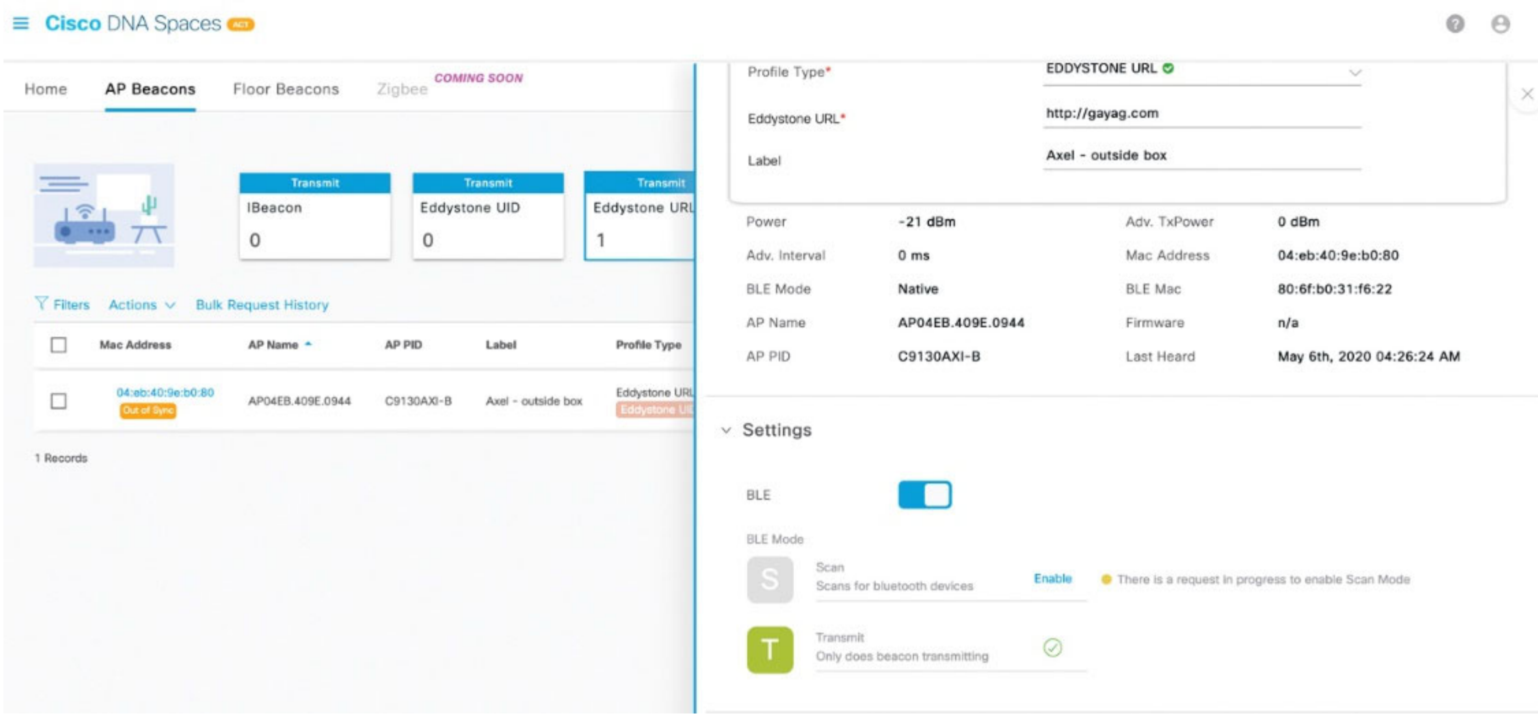


Figure 11-23 Cisco DNA Spaces allows you to choose the BLE mode of your AP

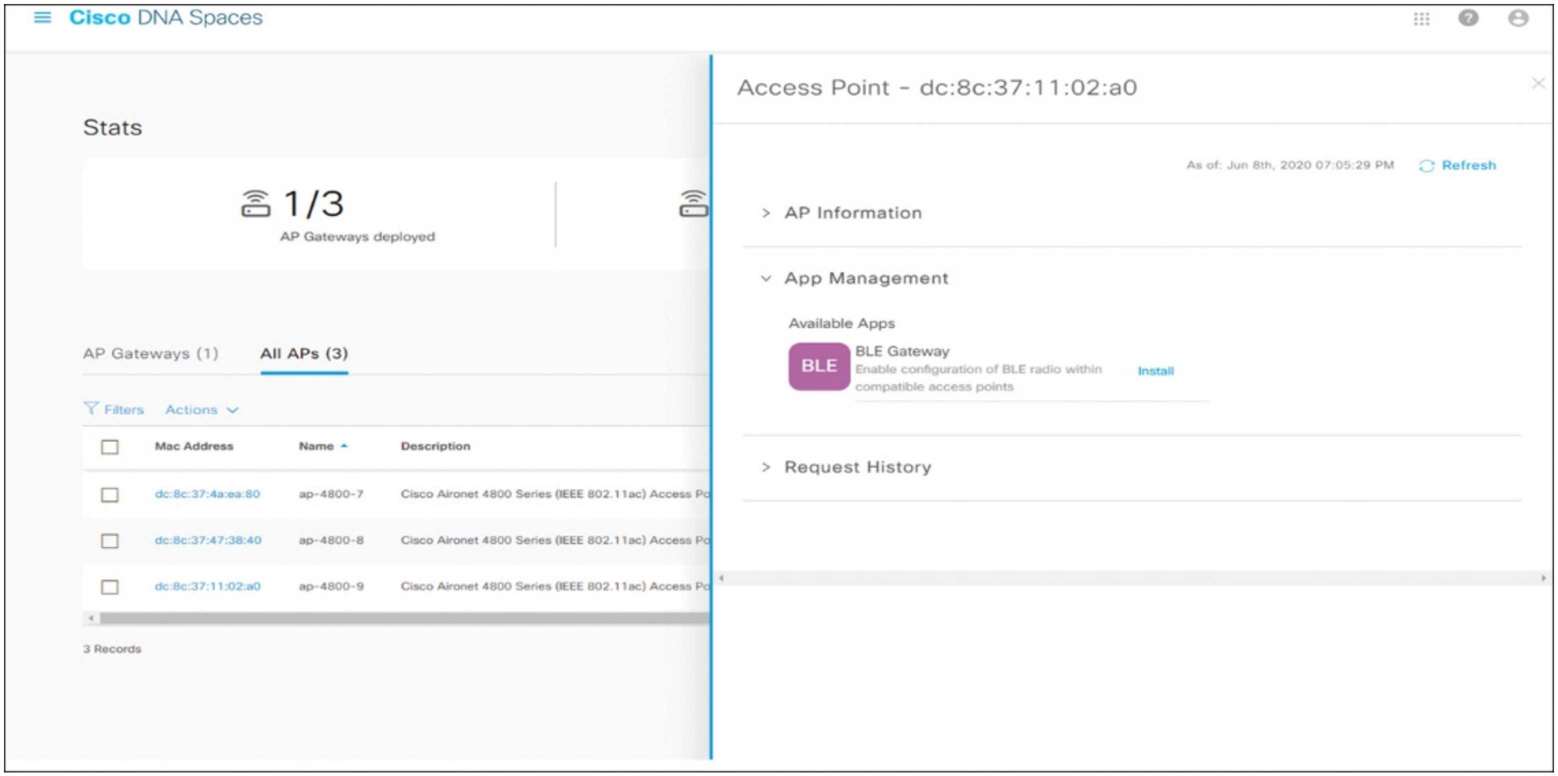


Figure 11-24 Access Point app management in Cisco DNA Spaces

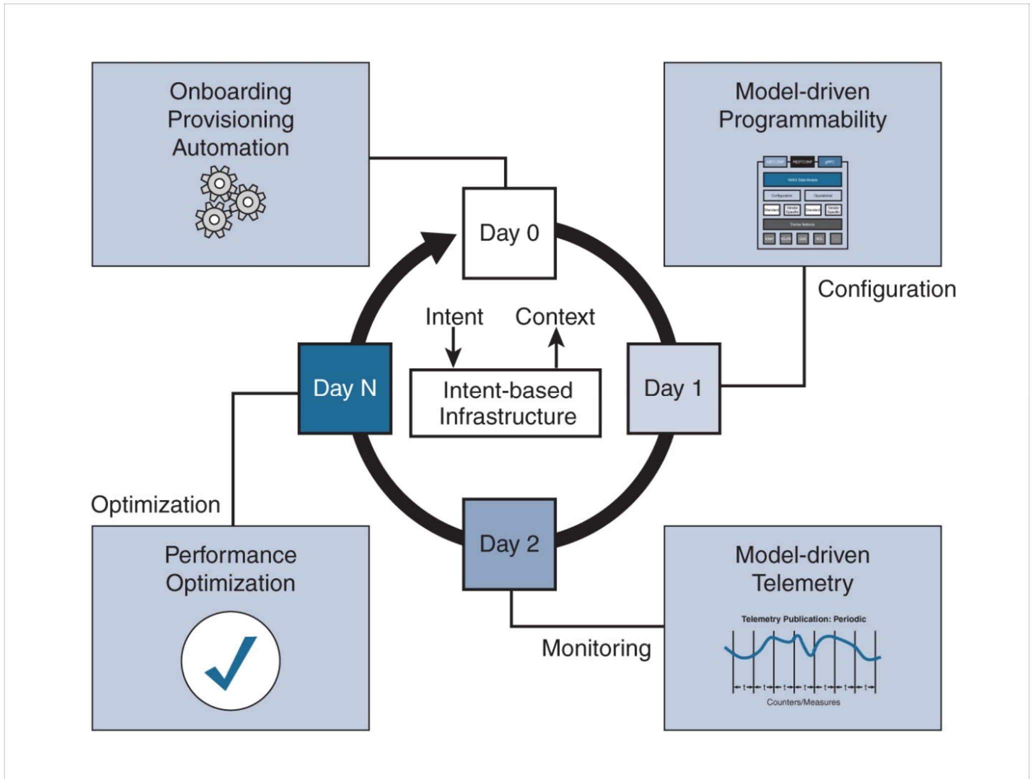
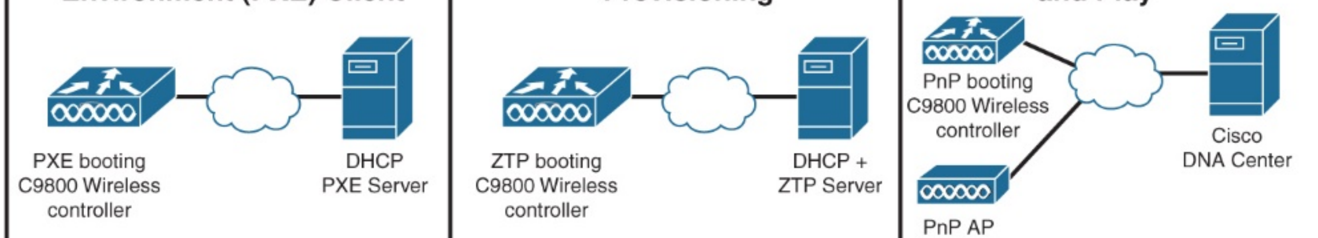


Figure 12-1 Programmability is used along all deployment phases



	Pre-boot Execution Environment (PXE) Client	Zero Touch Provisioning	Cisco Network Plug and Play
Image source	Network	Device	Device
Interfaces	Open/Standards based	Open/Standards based	"Turn-key" solution
Key Values	Ideal for heterogeneous/multi-Vendor network environments	Ideal for heterogeneous/multi-Vendor network environments	• Optimized for Cisco enterprise networks • Highly secure • Scalable

Figure 12-2 Day 0 protocols

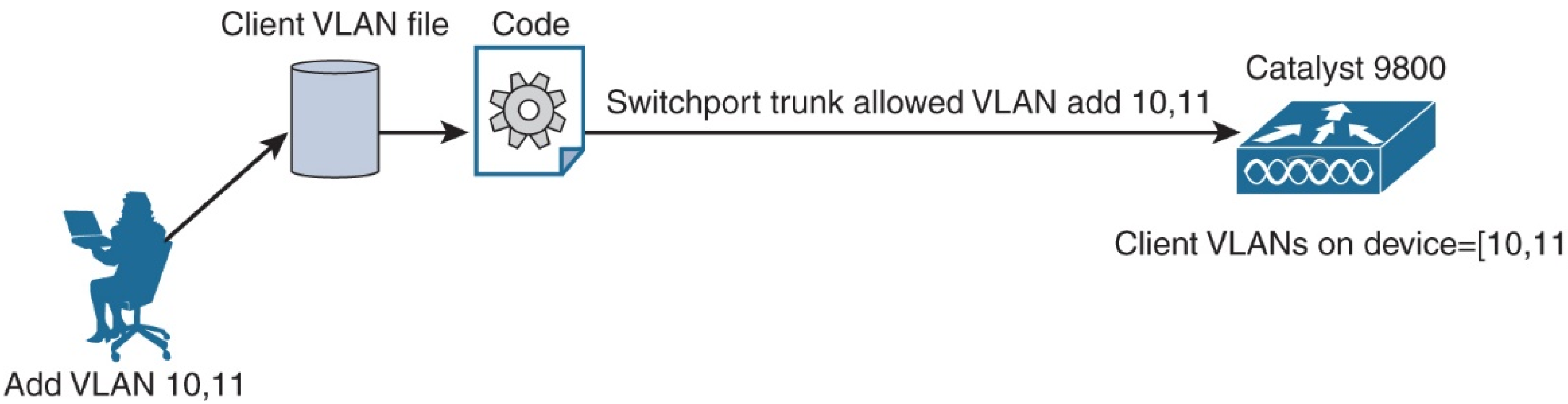


Figure 12-3 Adding client VLANs using the imperative model

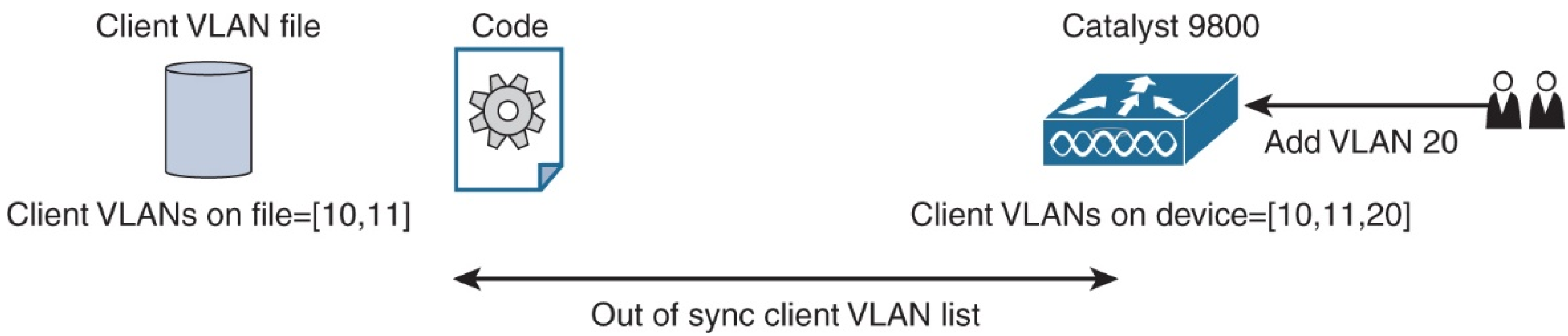


Figure 12-4 Inconsistency by using imperative models

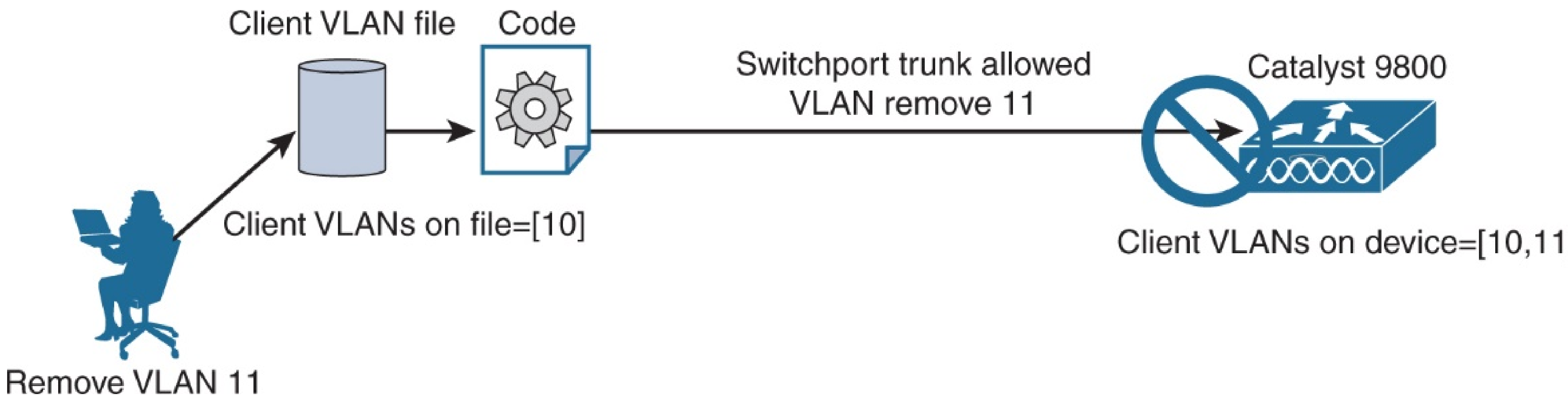


Figure 12-5 Errors when using imperative models

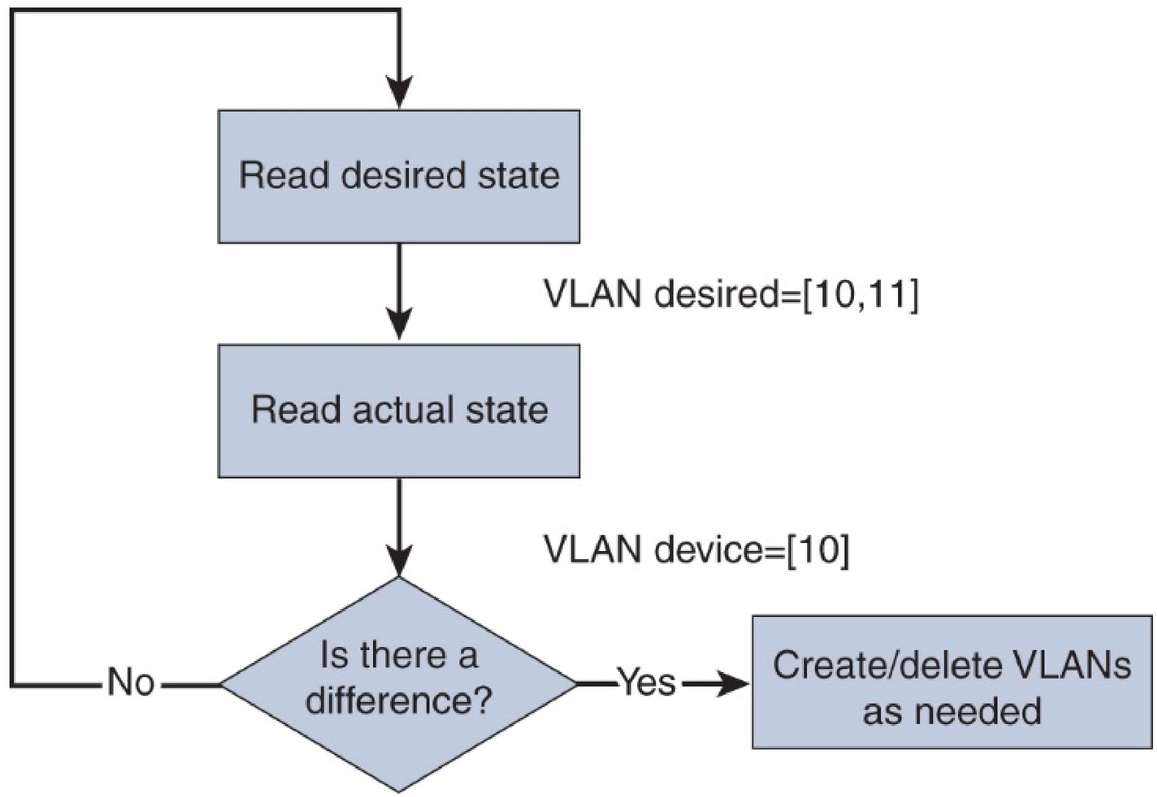


Figure 12-6 Declarative (closed-loop) model

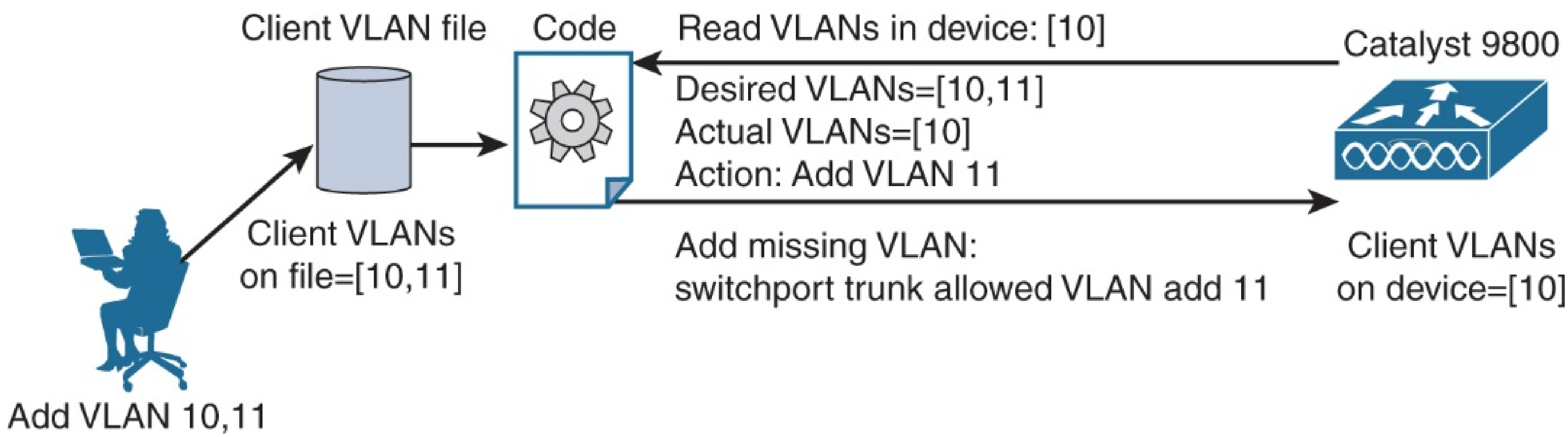


Figure 12-7 Updating data using declarative models

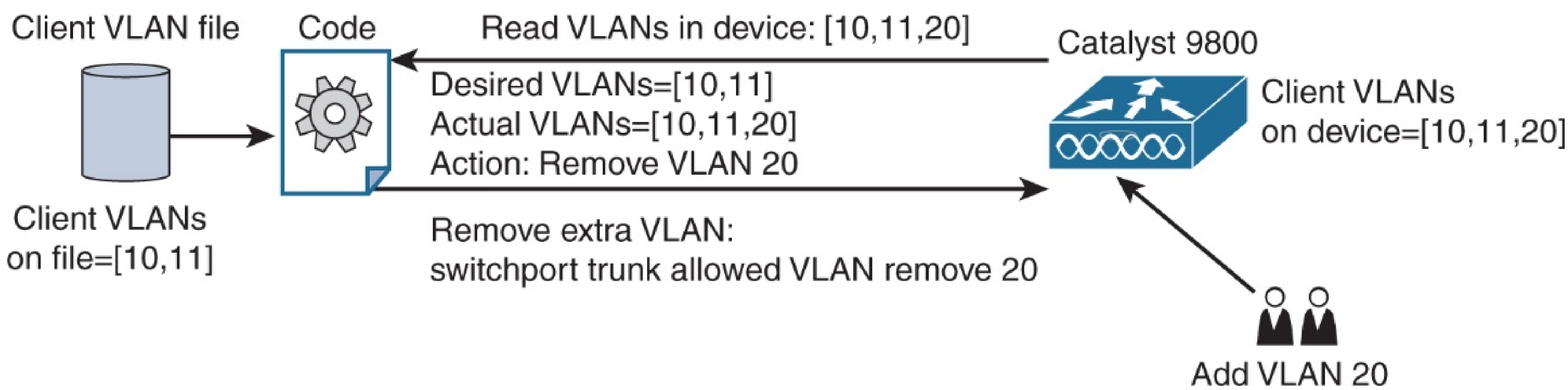


Figure 12-8 Automatic correction using declarative models

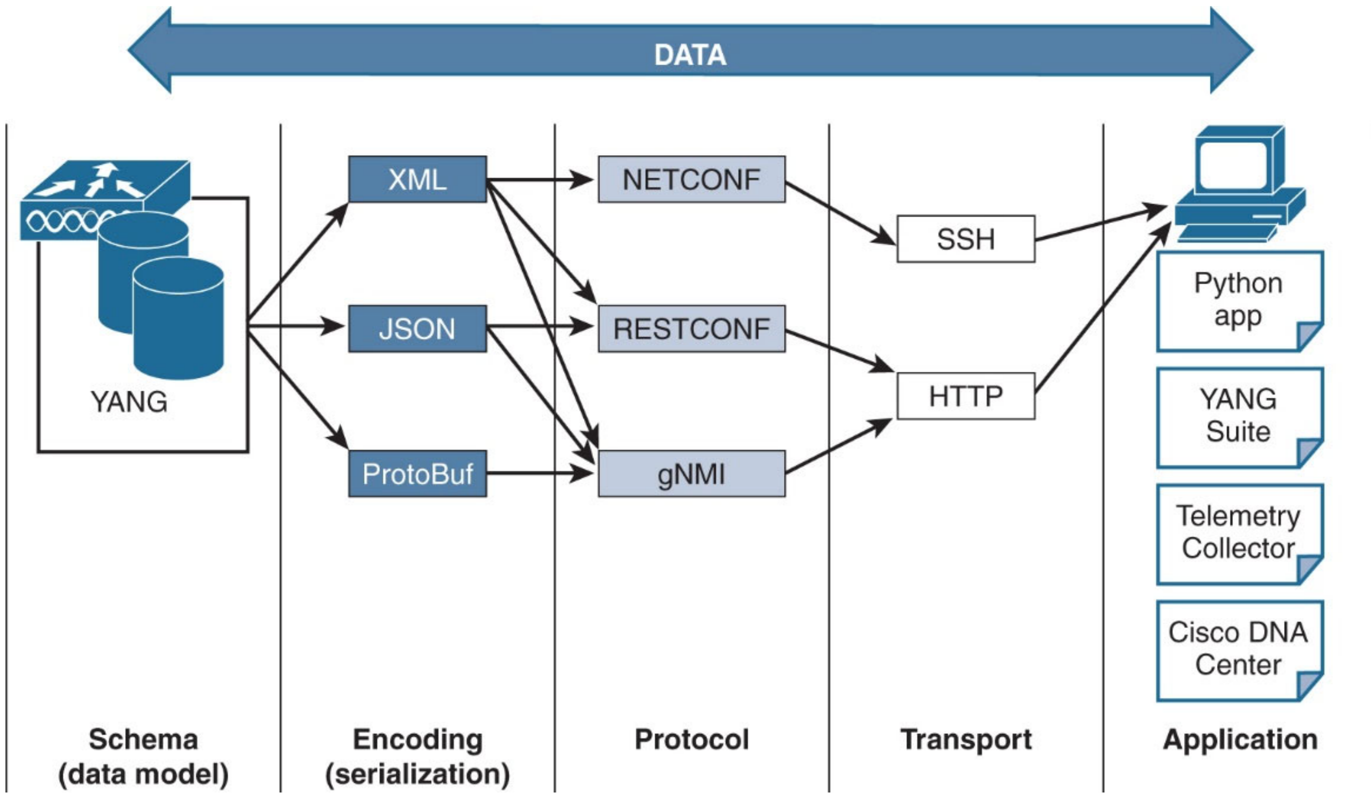


Figure 12-9 Protocols, encoding, and transport

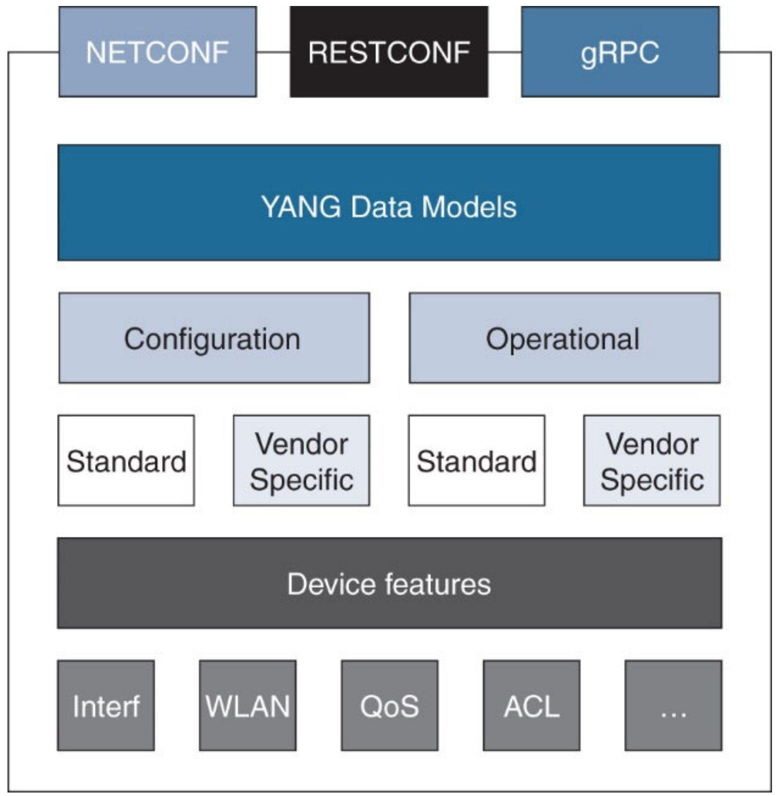


Figure 12-10 Data model and protocol hierarchy

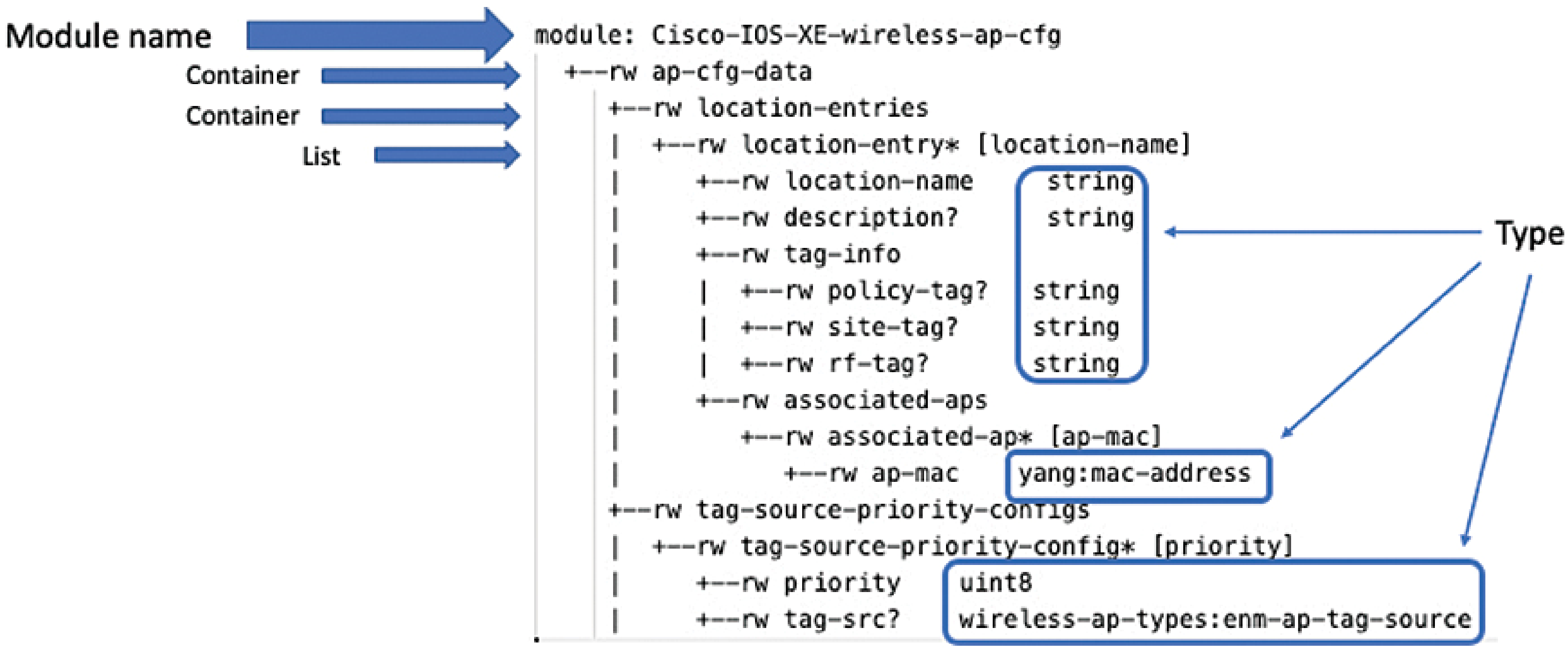


Figure 12-11 YANG module structure

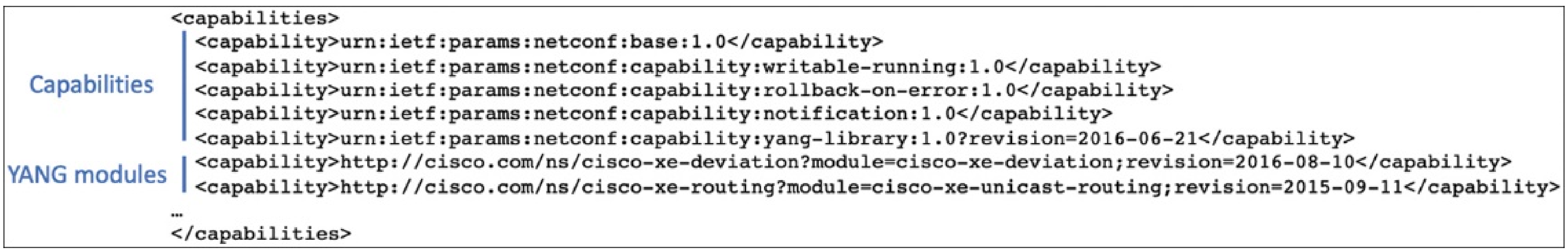


Figure 12-12 NETCONF capabilities

```
RFC 6241                    NETCONF Protocol                  June 2011

             Layer                 Example
        +-------------+      +-----------------+      +----------------+
    (4) |   Content   |      |  Configuration  |      |  Notification  |
        |             |      |      data       |      |      data      |
        +-------------+      +-----------------+      +----------------+
               |                       |                      |
        +-------------+      +-----------------+              |
    (3) | Operations  |      |  <edit-config>  |              |
        |             |      |                 |              |
        +-------------+      +-----------------+              |
               |                       |                      |
        +-------------+      +-----------------+      +----------------+
    (2) |  Messages   |      |     <rpc>,      |      | <notification> |
        |             |      |   <rpc-reply>   |      |                |
        +-------------+      +-----------------+      +----------------+
               |                       |                      |
        +-------------+      +-----------------------------------------+
    (1) |   Secure    |      |  SSH, TLS, BEEP/TLS, SOAP/HTTP/TLS, ... |
        |  Transport  |      |                                         |
        +-------------+      +-----------------------------------------+
```

Figure 12-13 NETCONF layers

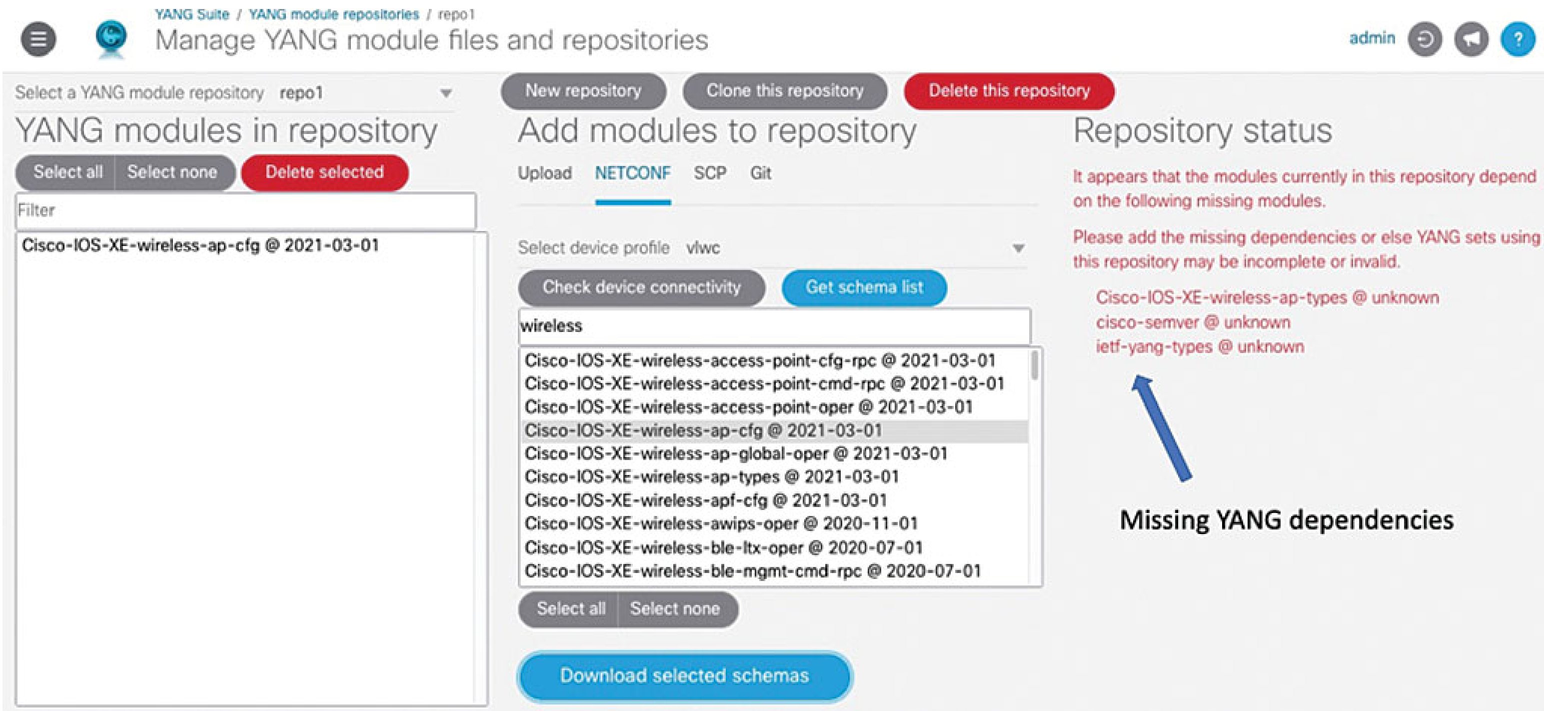


Figure 12-14 Missing dependencies in a YANG Suite repository

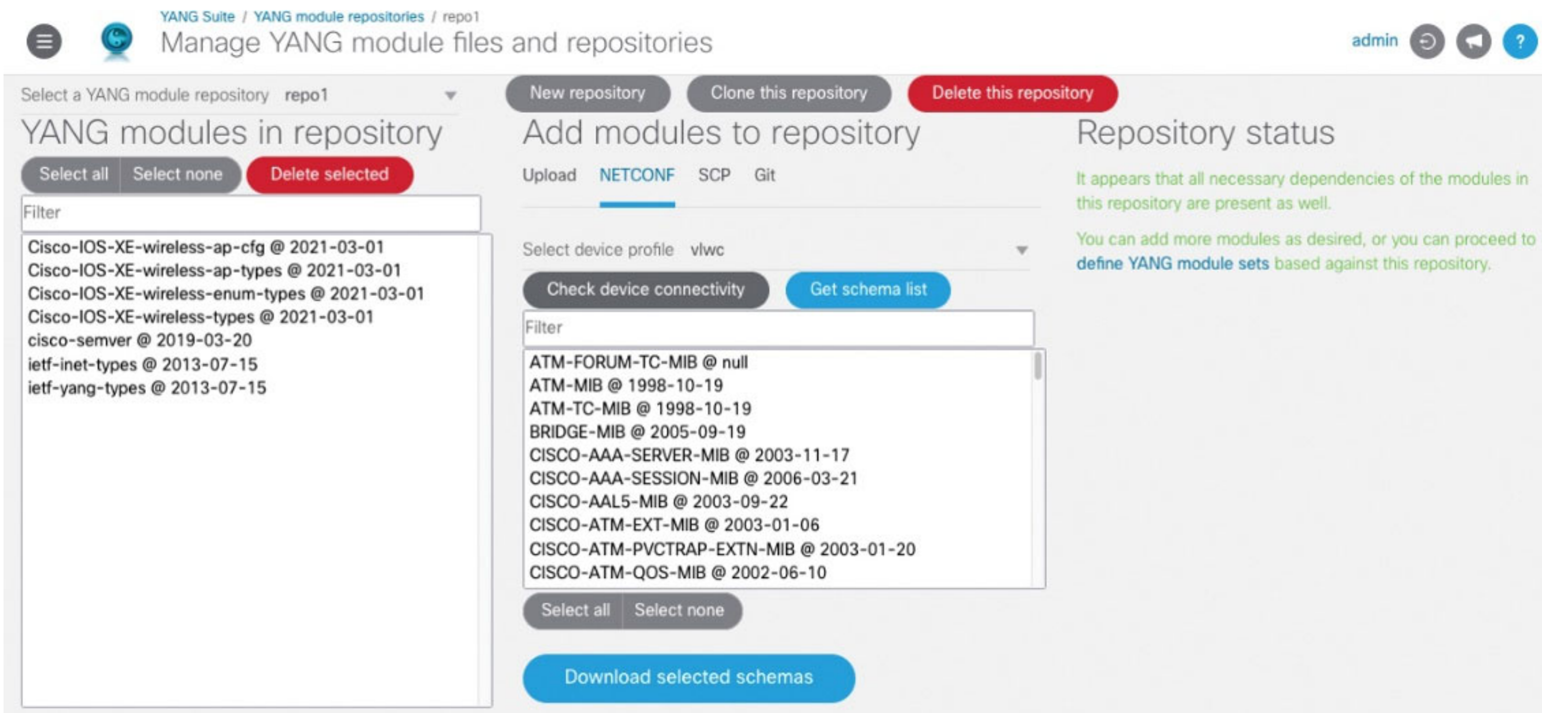


Figure 12-15 YANG Suite repository ready to be used

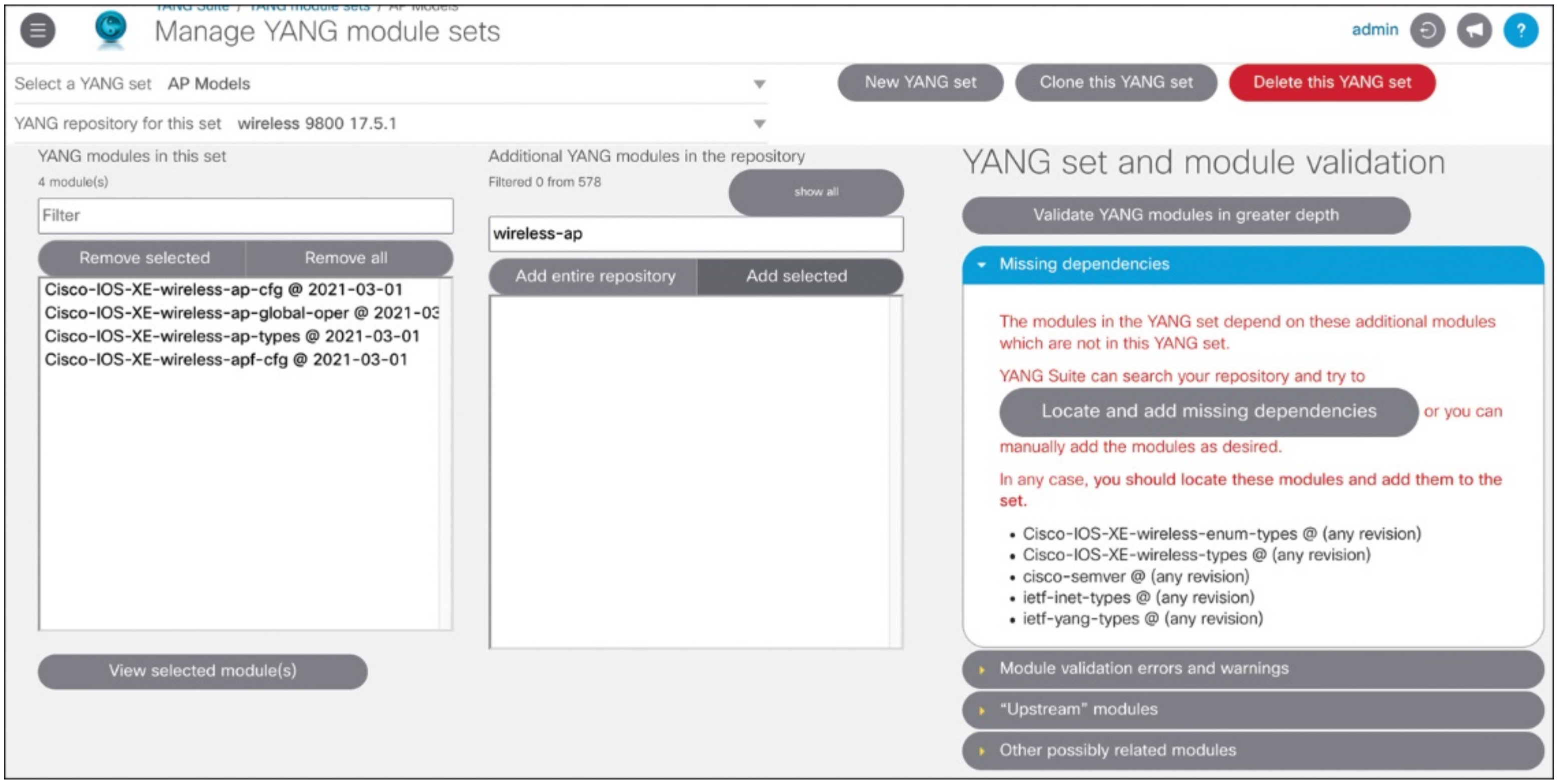


Figure 12-16 YANG Suite displaying missing dependencies

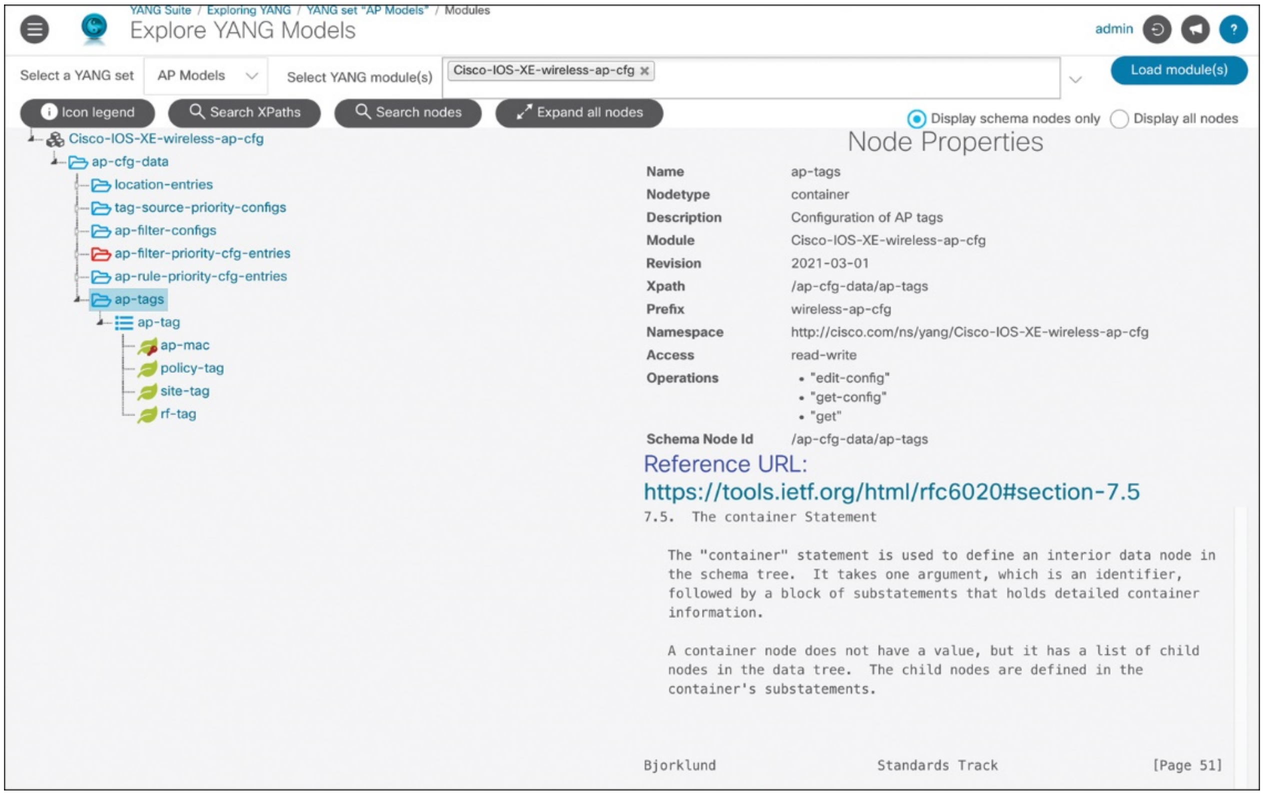


Figure 12-17 Exploring YANG models

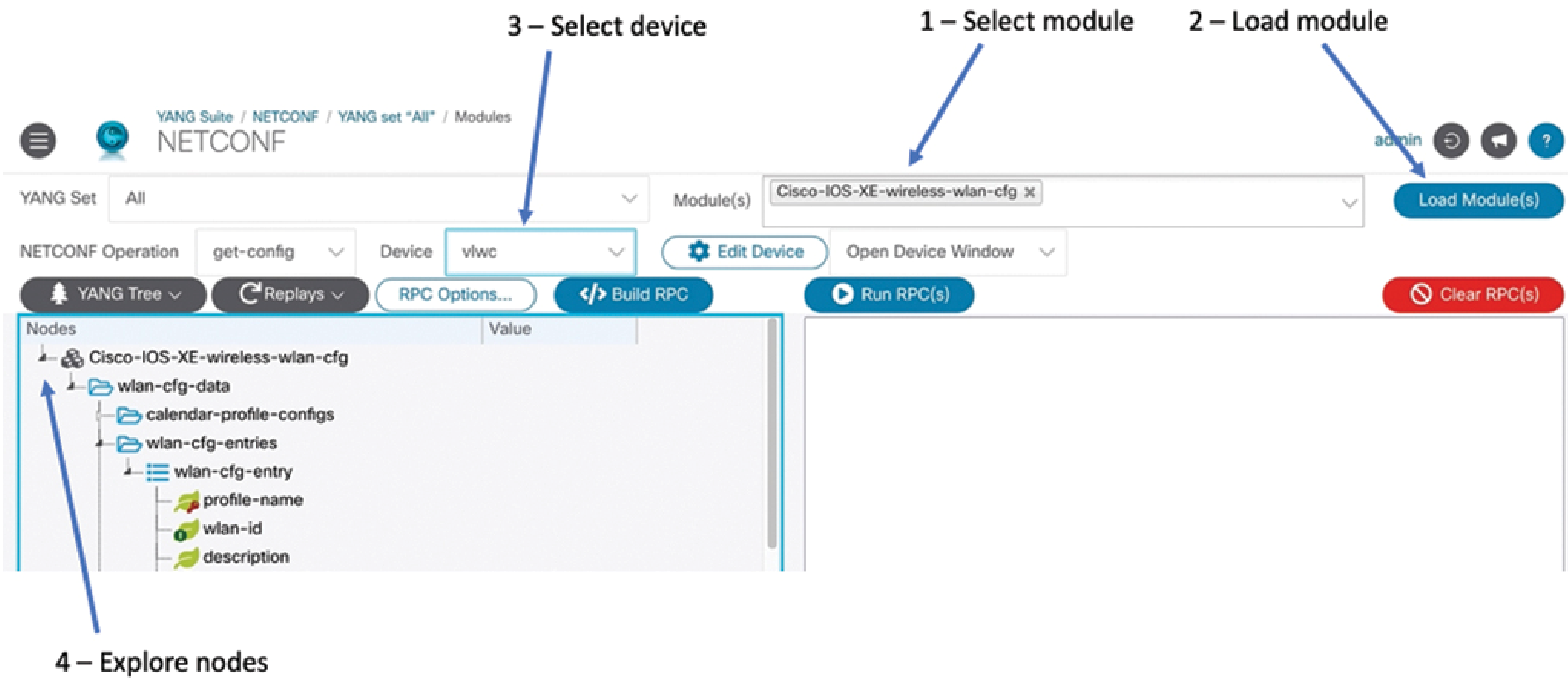


Figure 12-18 Exploring data in YANG Suite

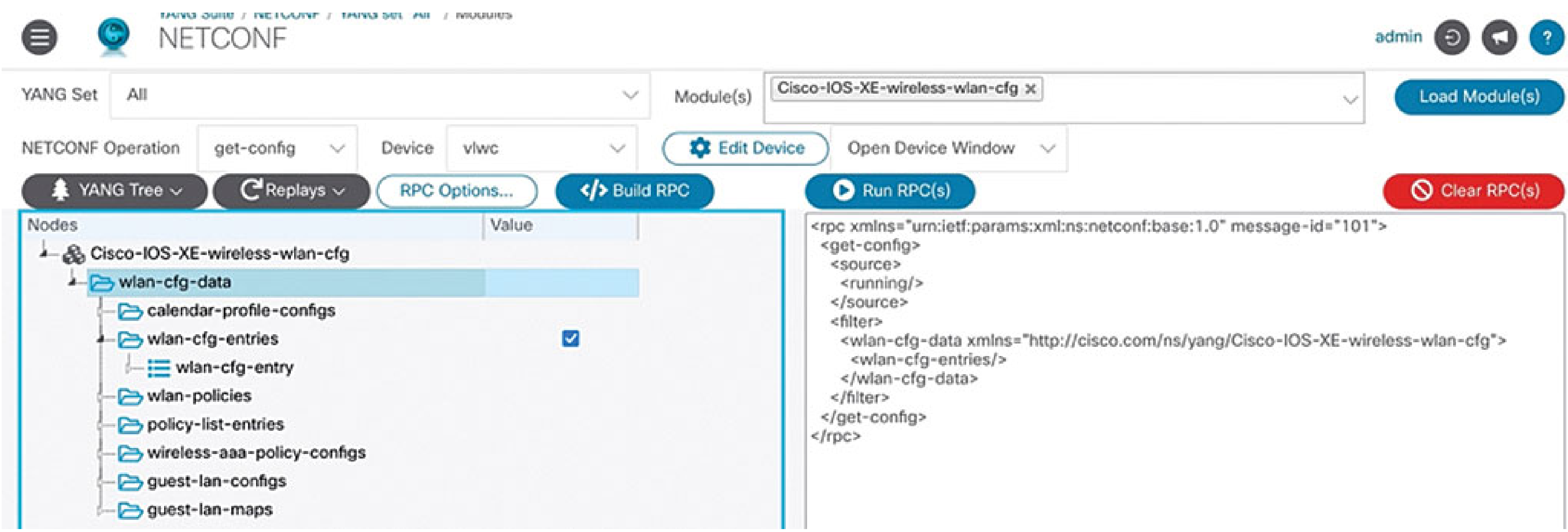


Figure 12-19 Building RPCs in YANG Suite

```
1  <nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="…">
2    <nc:get-config>
       <nc:source>
3        <nc:running/>
       </nc:source>
       <nc:filter>
4        <wlan-cfg-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-wlan-cfg">
5          <wlan-cfg-entries/>
         </wlan-cfg-data>
       </nc:filter>
     </nc:get-config>
   </nc:rpc>
```

Figure 12-20 NETCONF RPC request

```
1  <rpc-reply message-id="..." xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
     <data>
2     <wlan-cfg-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-wlan-cfg">
3      <wlan-cfg-entries>
4       <wlan-cfg-entry>
         <profile-name>test_wlan_1</profile-name>
         <wlan-id>1</wlan-id>
         <wep-key-index>1</wep-key-index>
         <multicast-buffer-value>0</multicast-buffer-value>
         <apf-vap-id-data>
          <ssid>test_wlan_1</ssid>
         </apf-vap-id-data>
        </wlan-cfg-entry>
5       <wlan-cfg-entry>
         <profile-name>test_wlan_2</profile-name>
         <wlan-id>2</wlan-id>
         <wep-key-index>1</wep-key-index>
         <multicast-buffer-value>0</multicast-buffer-value>
         <apf-vap-id-data>
          <ssid>test_wlan_2</ssid>
          <wlan-status>true</wlan-status>
         </apf-vap-id-data>
        </wlan-cfg-entry>
       </wlan-cfg-entries>
      </wlan-cfg-data>
     </data>
    </rpc-reply>
```

Figure 12-21 NETCONF RPC reply

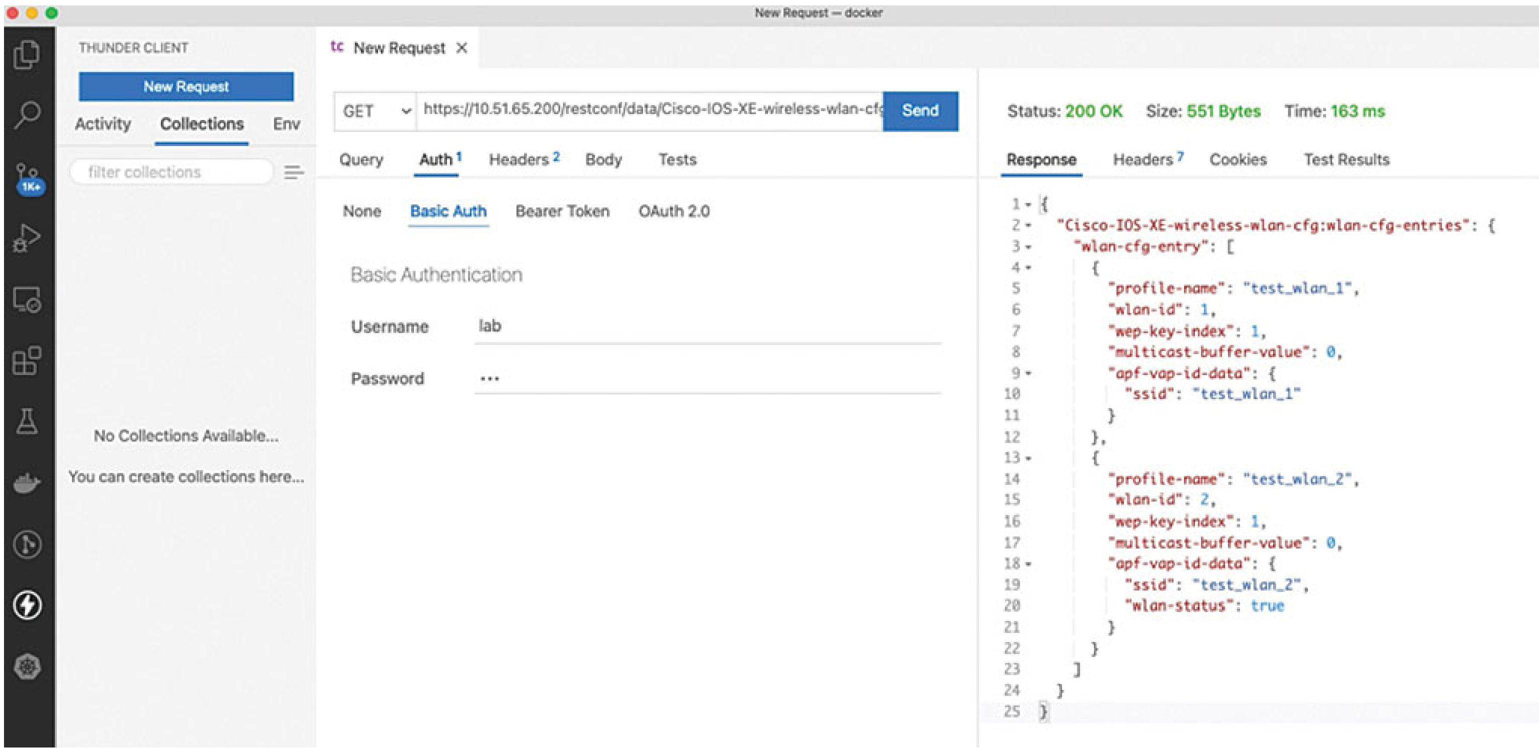


Figure 12-22 Thunder plug-in for Visual Studio Code

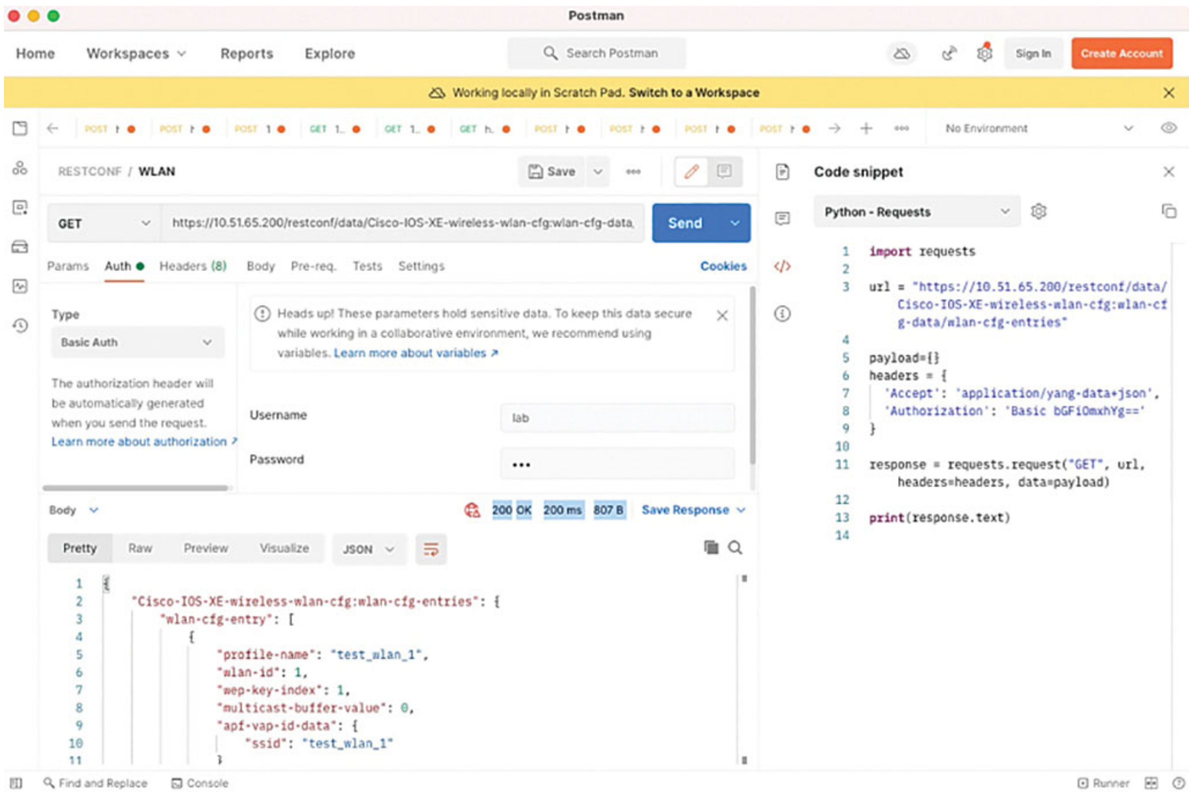


Figure 12-23 Postman

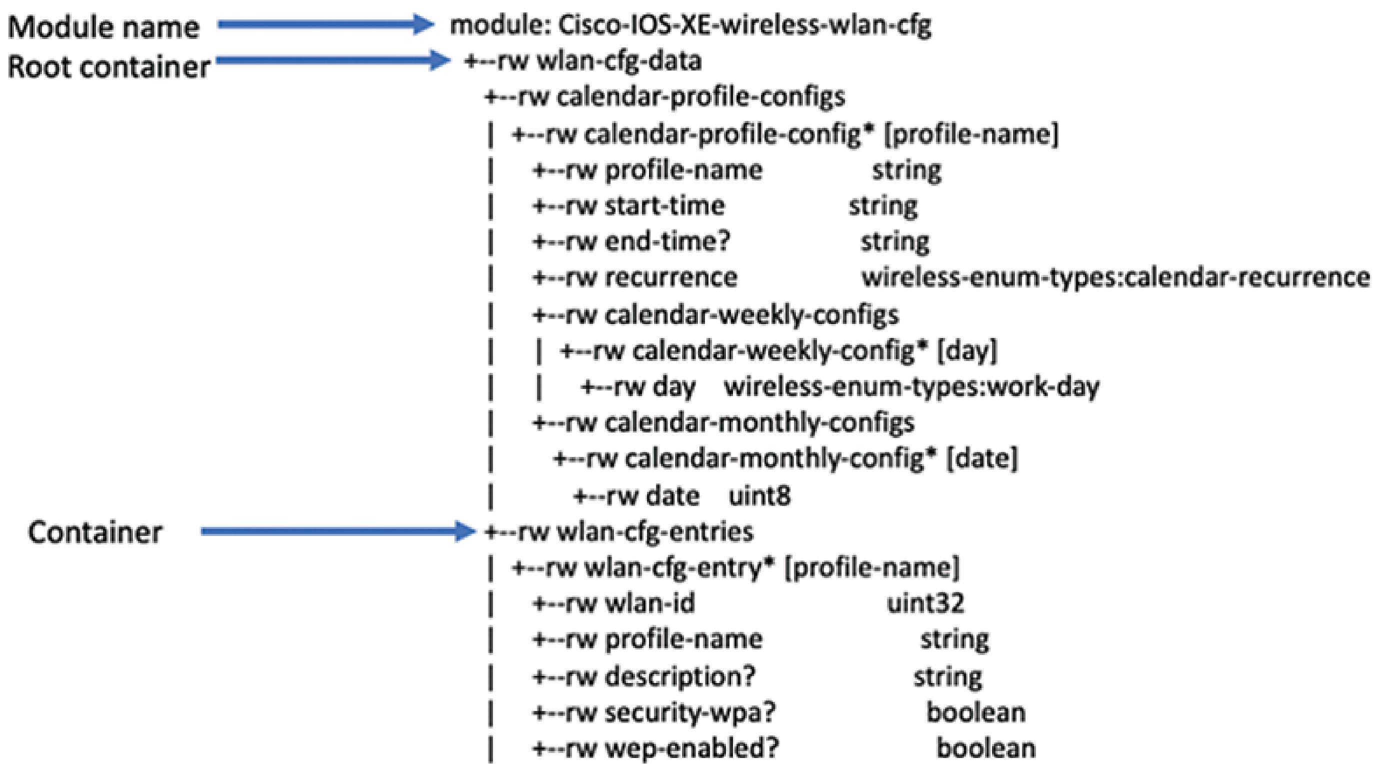


Figure 12-24 WLAN YANG Model

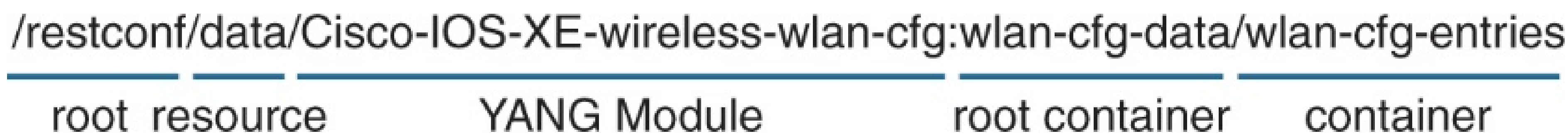


Figure 12-25 Constructing URI from YANG data model

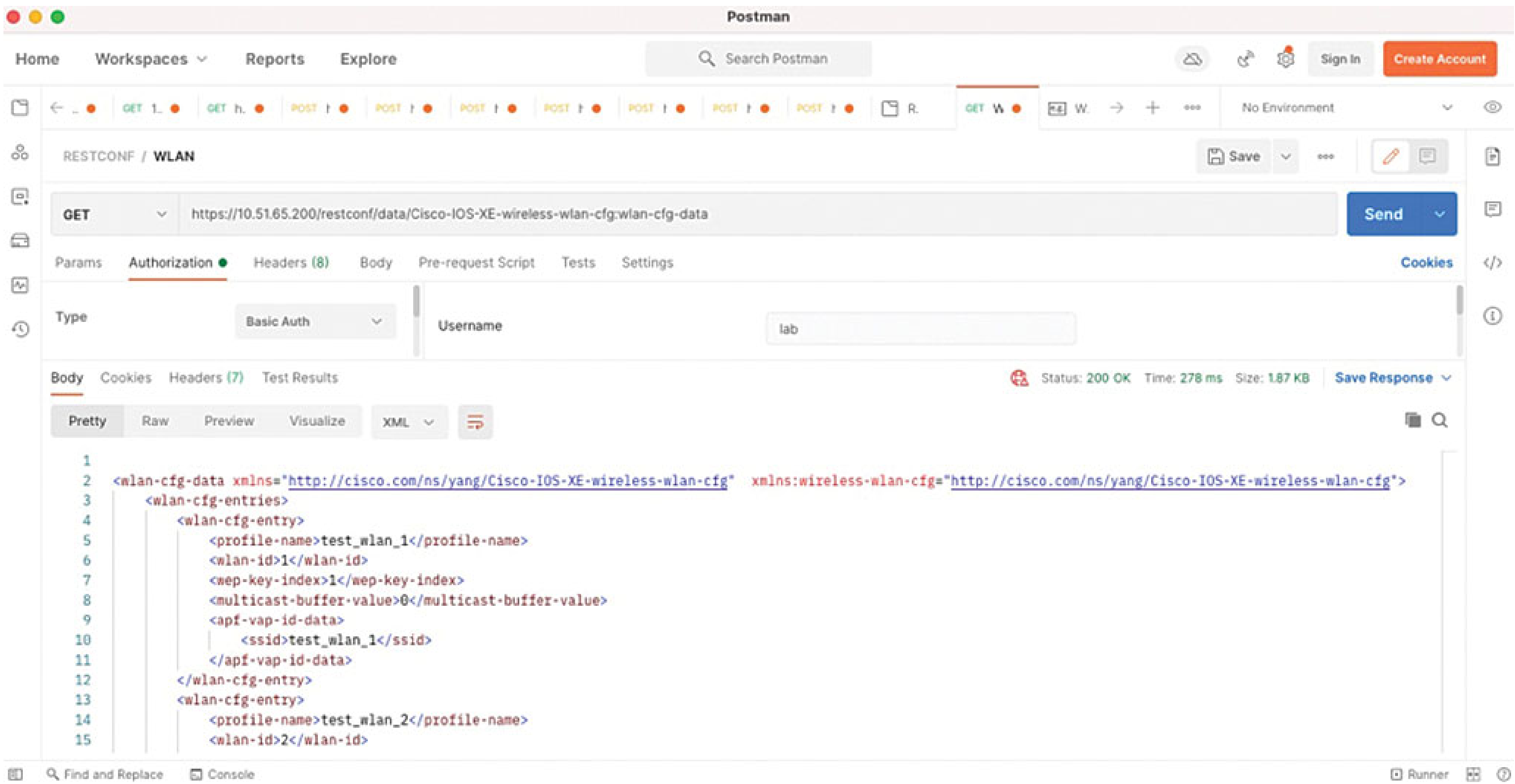


Figure 12-26 Using Postman to retrieve a list of WLANs

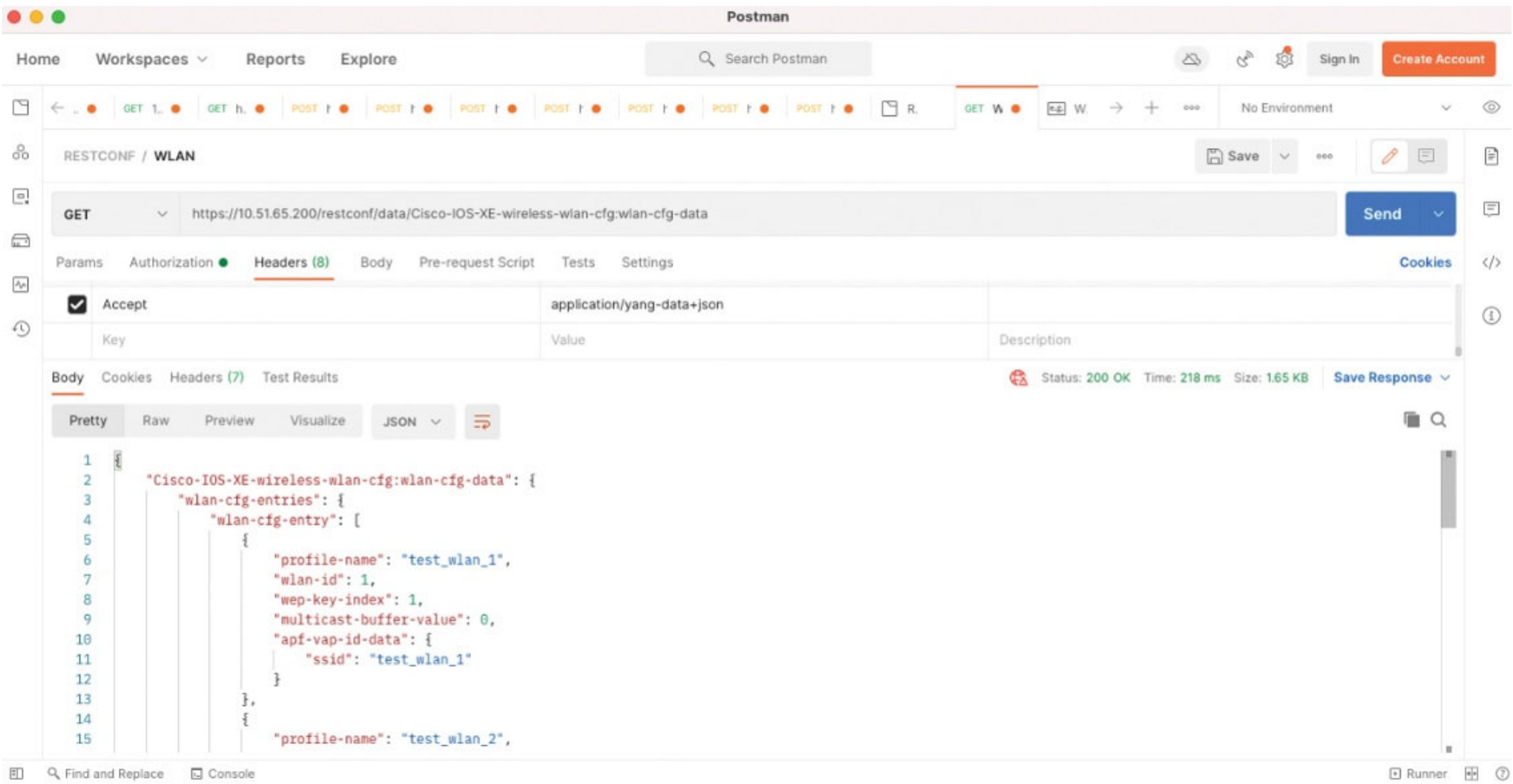


Figure 12-27 Requesting JSON data in Postman

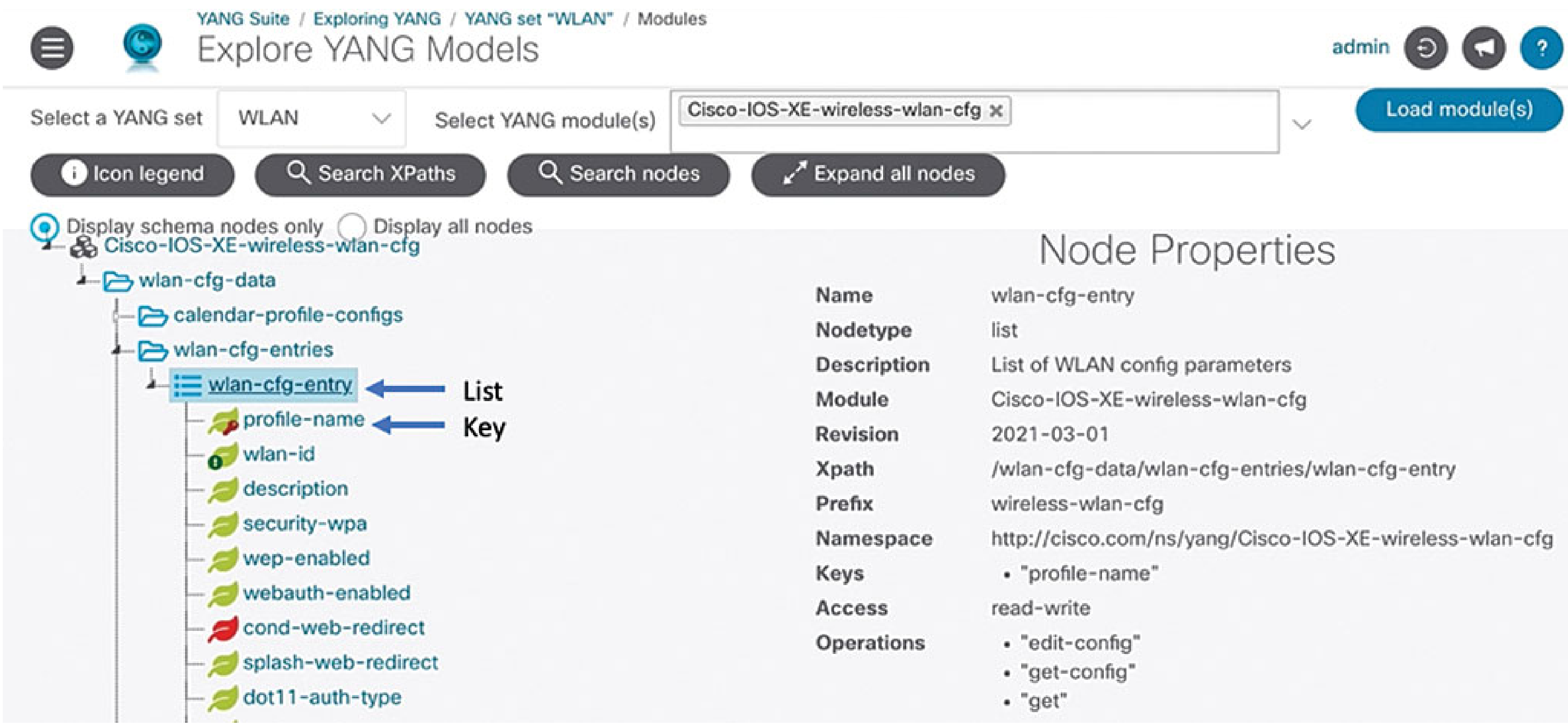


Figure 12-28 List and key in the WLAN config entry model

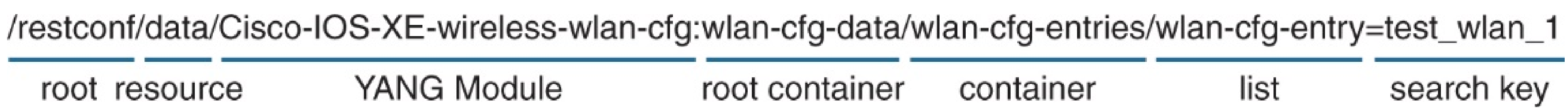


Figure 12-29 Constructing searches in RESTCONF

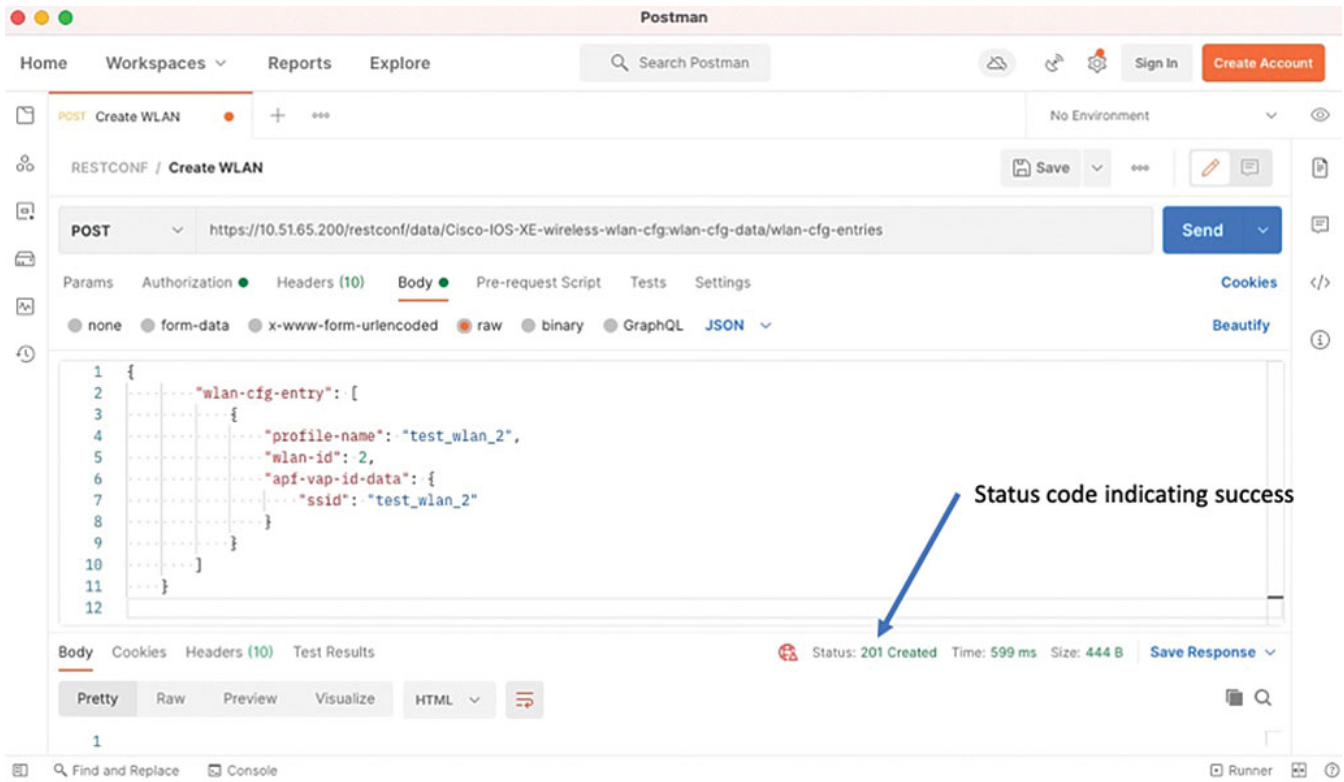


Figure 12-30 Adding a new WLAN using Postman

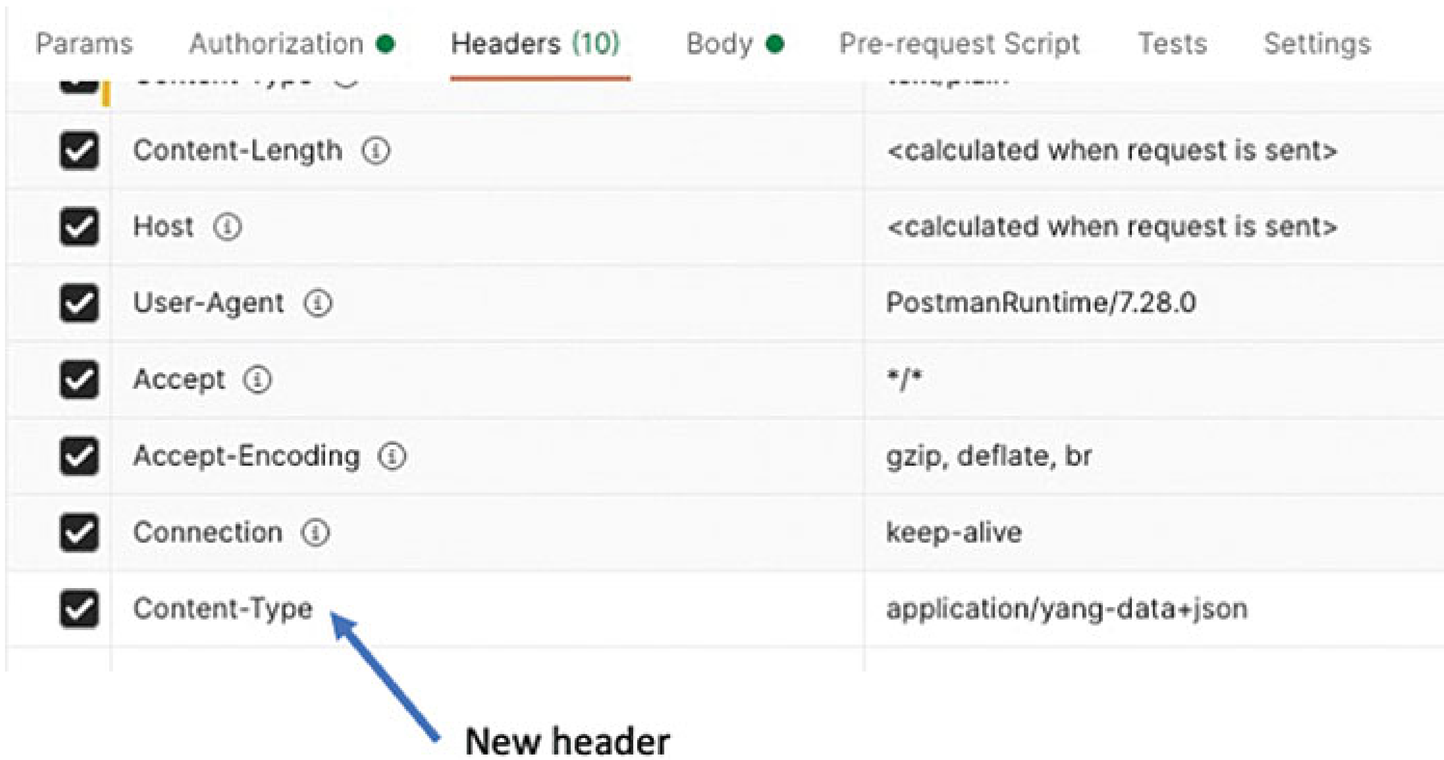


Figure 12-31 Content-type header

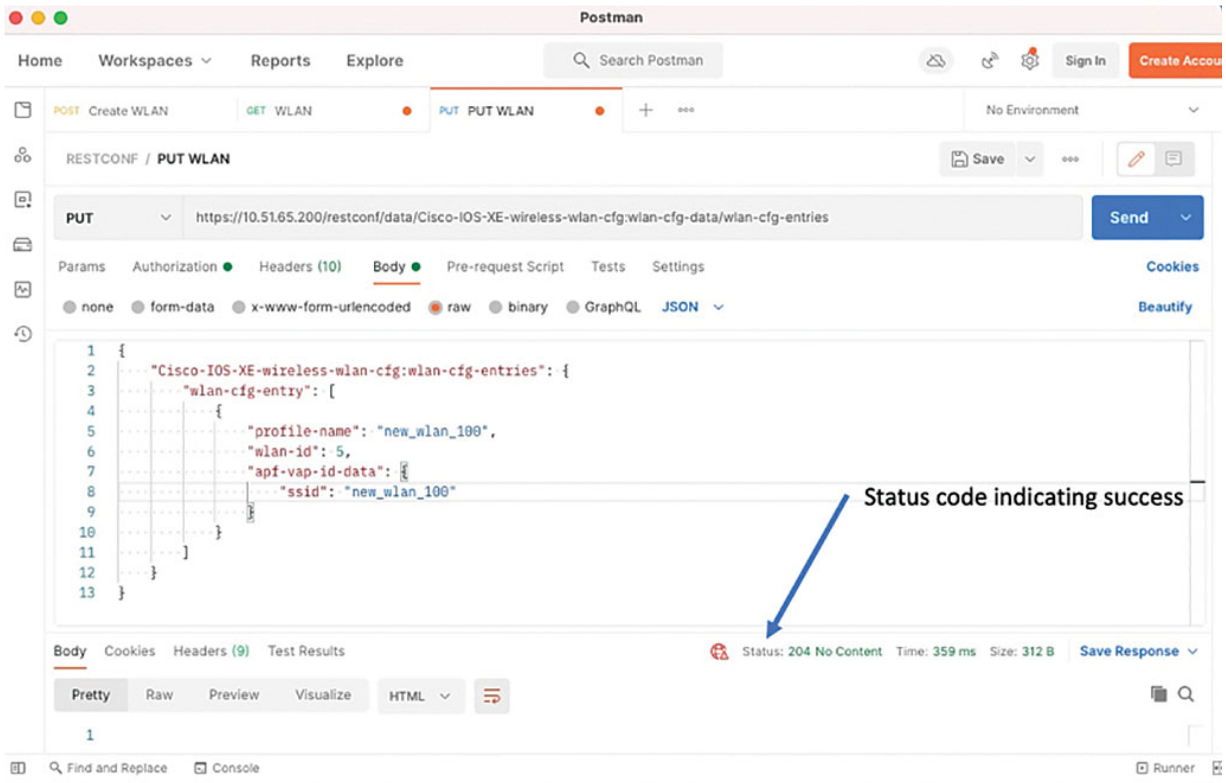


Figure 12-32 Replacing WLAN configuration using Postman

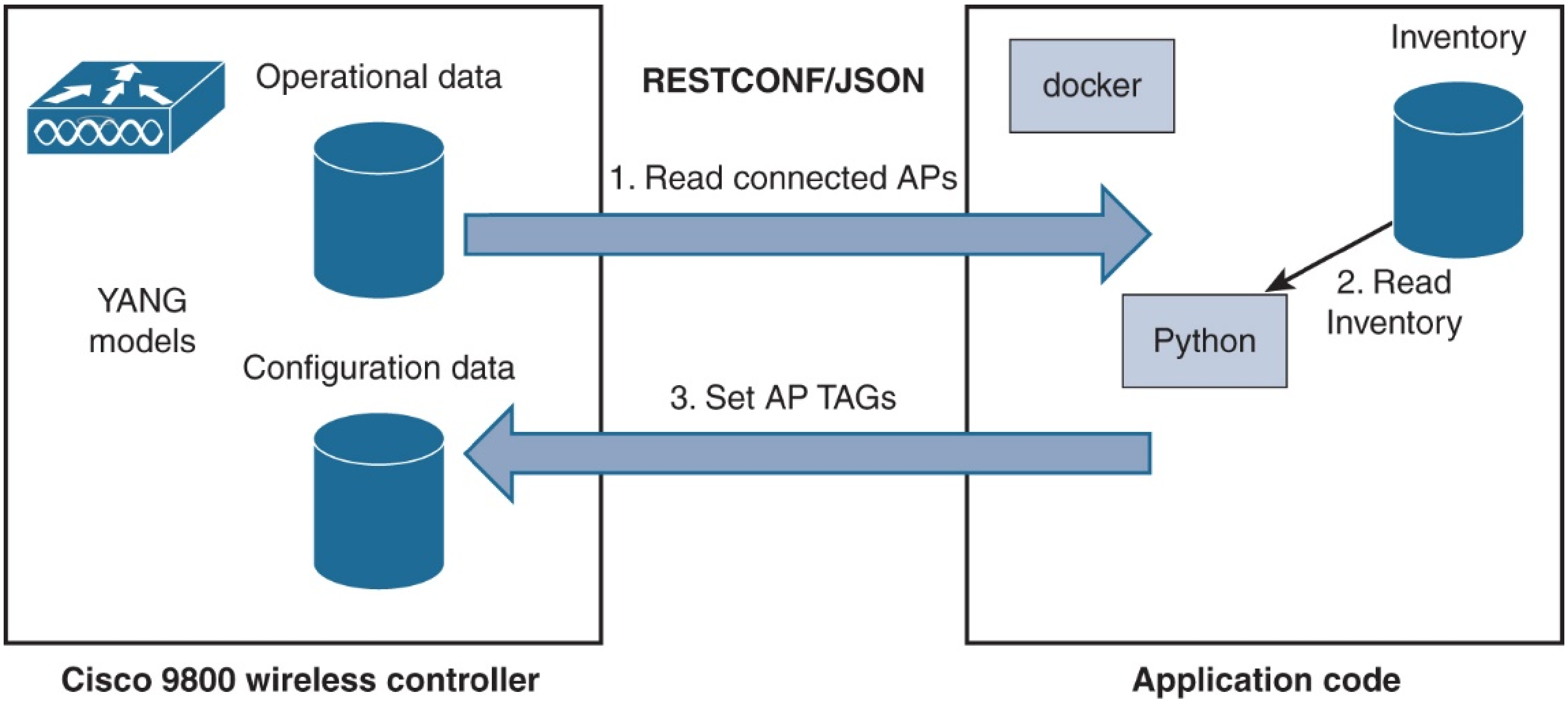


Figure 12-33 Python application to assign a tag based on the AP serial number

Figure 12-34 Inventory file

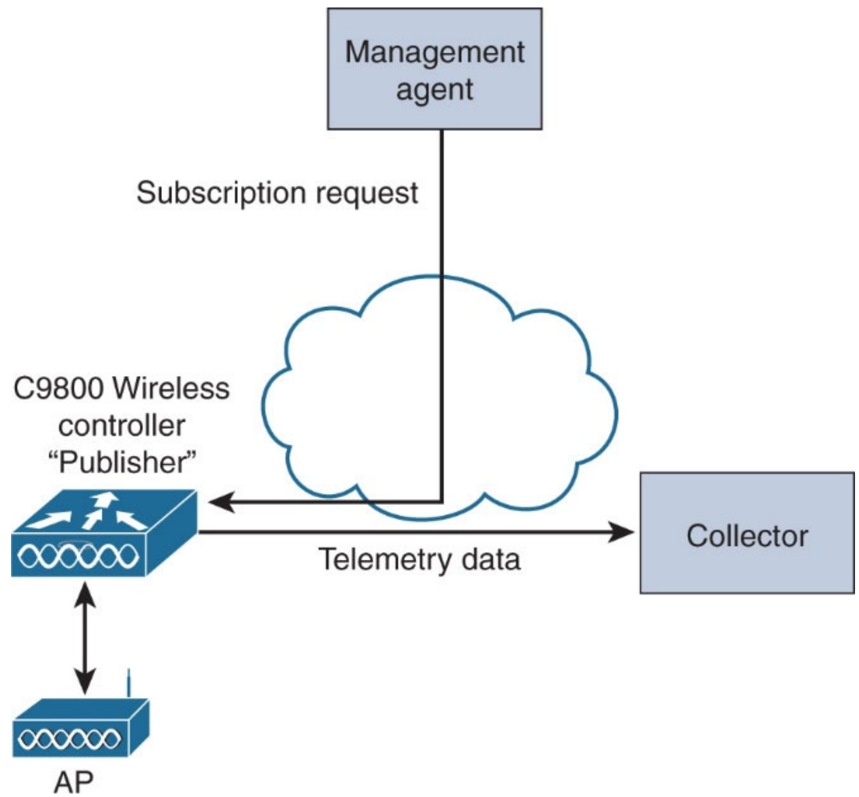


Figure 13-1 Roles in model-driven telemetry

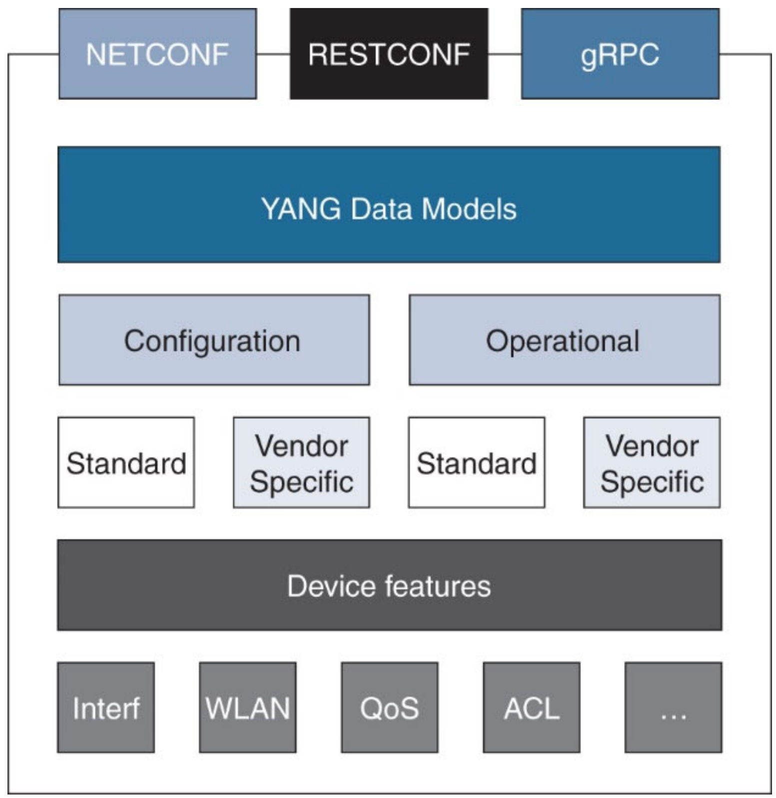


Figure 13-2 Configuration and operational YANG data models

```
container ap-global-oper-data {
  config false;
  description
    "Root container for AP operational data aggregated across wireless processes";
  container ap-img-predownload-stats {
    presence "ap-img-predownload-stats";
    description
      "AP image predownload stats";
    uses wireless-ap-global-oper:global-ap-stats;
  }
  list ap-join-stats {
    key "wtp-mac";
    description
      "AP join statistics";
    uses wireless-ap-global-oper:st-emltd-ap-stats-info;
  }
}
}
```

Indicates this is an operational model

Figure 13-3 Identifying an operational YANG model

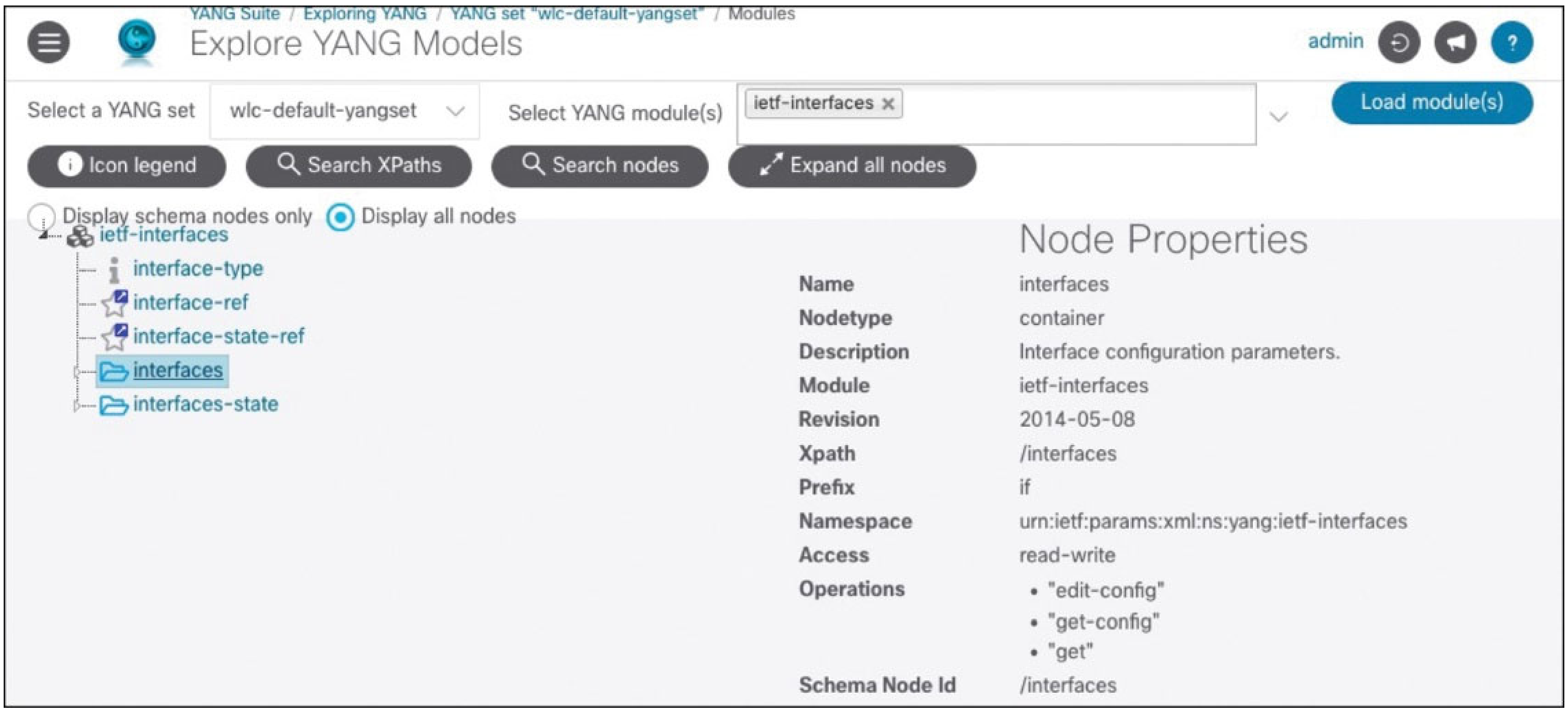


Figure 13-4 Configuration container for interfaces

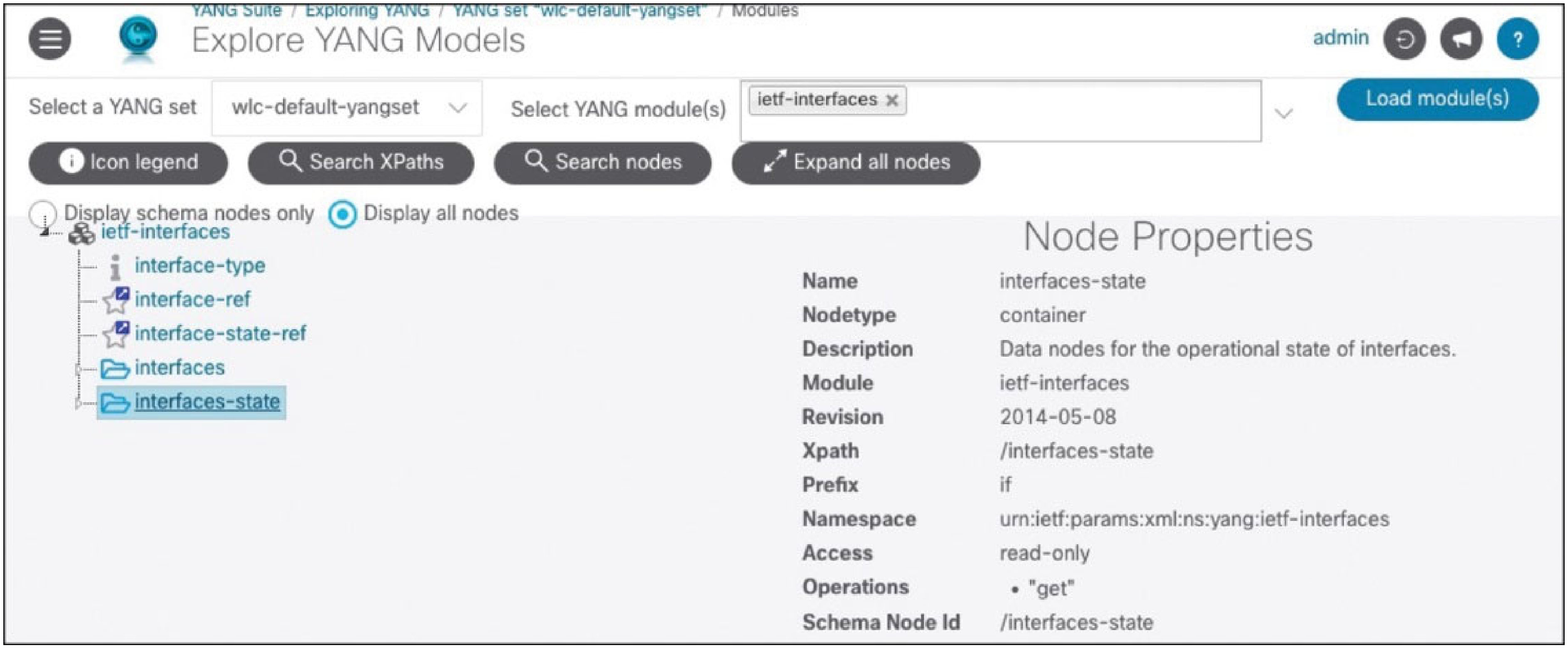


Figure 13-5 Operational container for interfaces-state

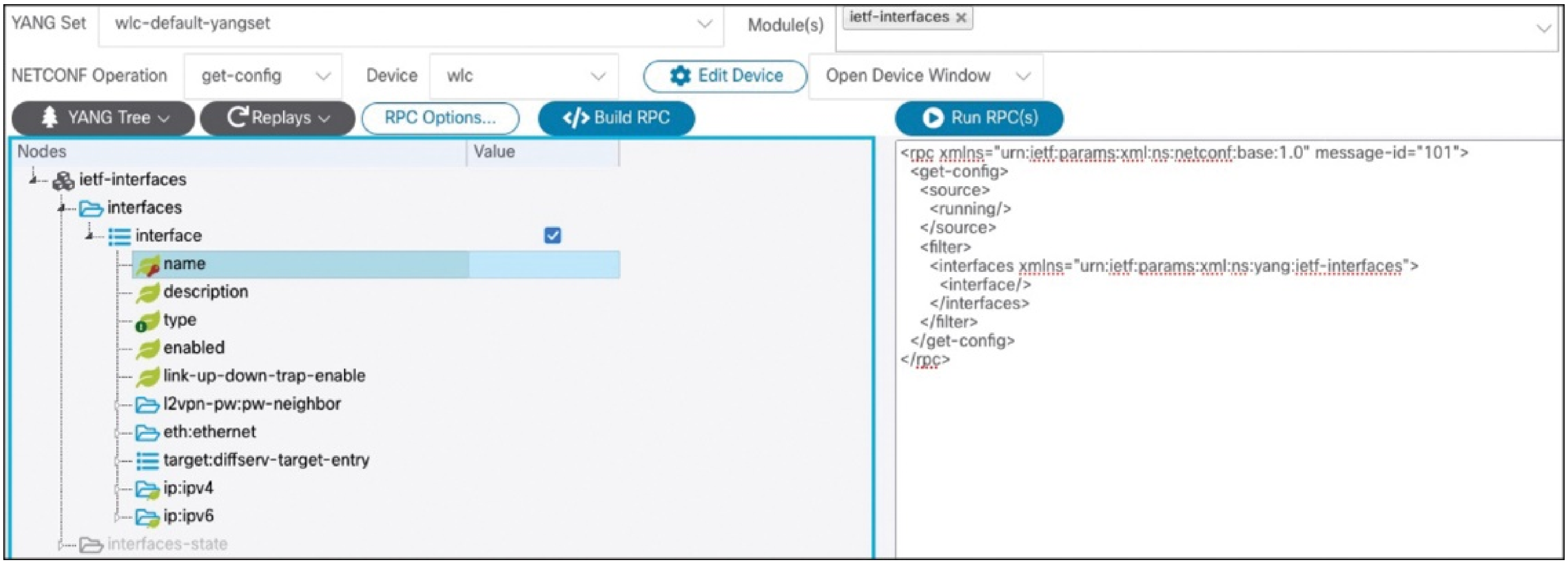


Figure 13-6 RPC to read configured interfaces using NETCONF

```
                              <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
get-config operation            <get-config>
                                  <source>
                                    <running/>
                                  </source>
                                  <filter>
   interfaces container             <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
                                      <interface/>
                                    </interfaces>
                                  </filter>
                                </get-config>
                              </rpc>
```

Figure 13-7 NETCONF query RPC to get interface list

GET operation

interfaces-state container

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <get>
    <filter>
      <interfaces-state xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name/>
          <oper-status/>
        </interface>
      </interfaces-state>
    </filter>
  </get>
</rpc>
```

Figure 13-8 NETCONF RPC to read the interfaces operational state

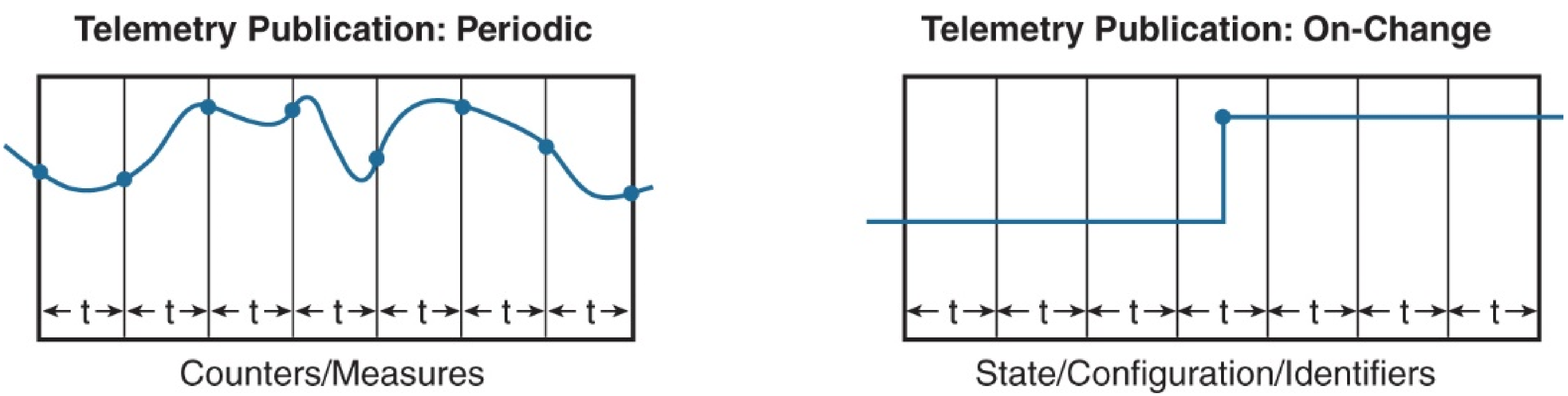
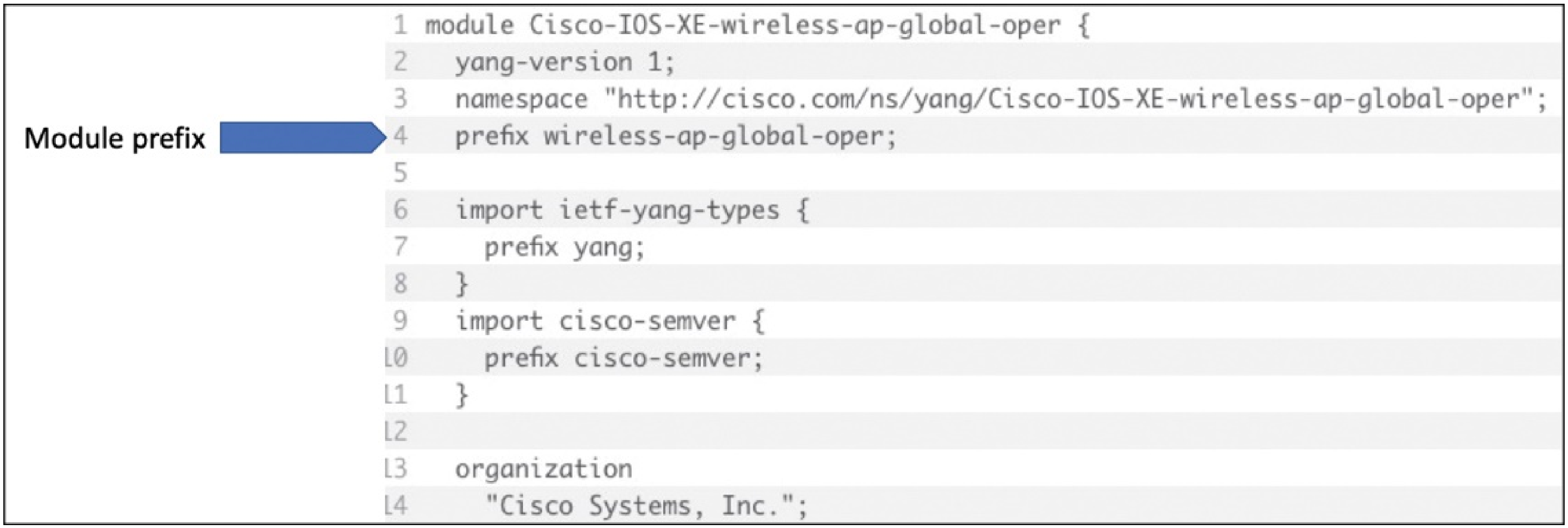


Figure 13-9 Periodic versus on-change publications




```
module Cisco-IOS-XE-wireless-ap-global-oper {
  yang-version 1;
  namespace "http://cisco.com/ns/yang/Cisco-IOS-XE-wireless-ap-global-oper";
  prefix wireless-ap-global-oper;

  import ietf-yang-types {
    prefix yang;
  }
  import cisco-semver {
    prefix cisco-semver;
  }

  organization
    "Cisco Systems, Inc.";
```

Figure 13-10 YANG module prefix in the YANG file

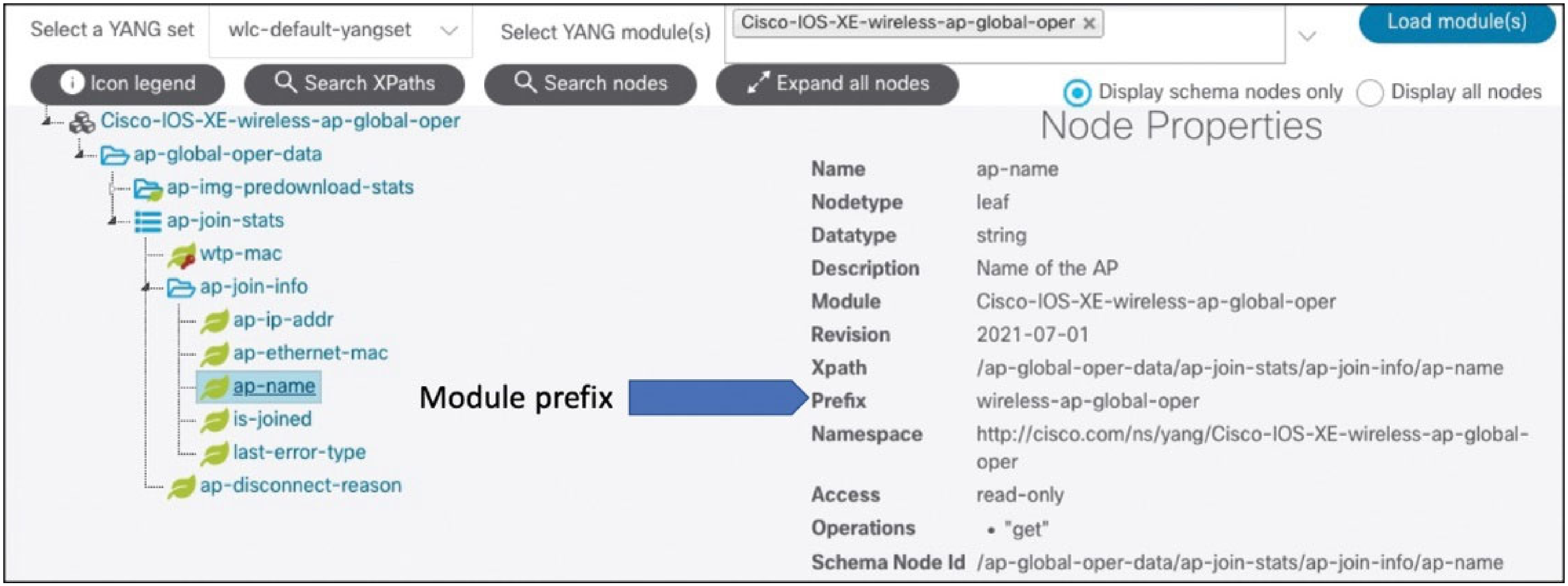


Figure 13-11 YANG module prefix as shown in YANG Suite

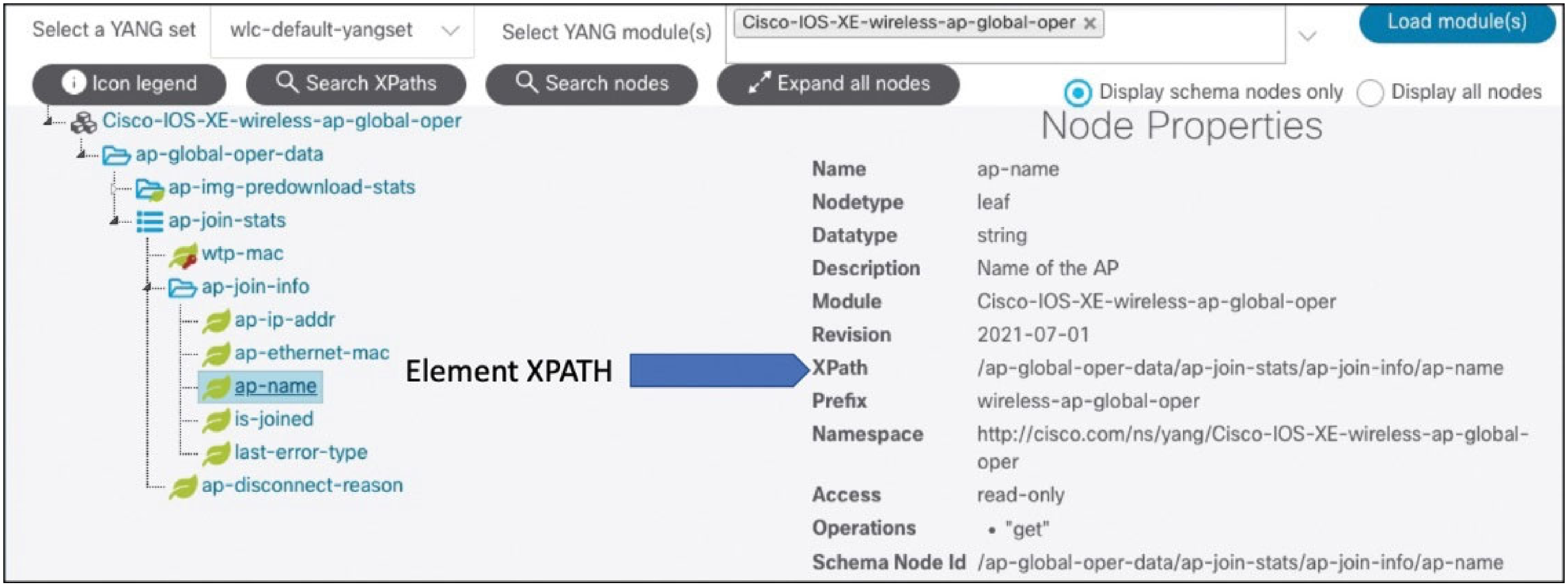


Figure 13-12 Element XPath as shown in YANG Suite

Figure 13-13 Composing a full XPath from the information in YANG Suite

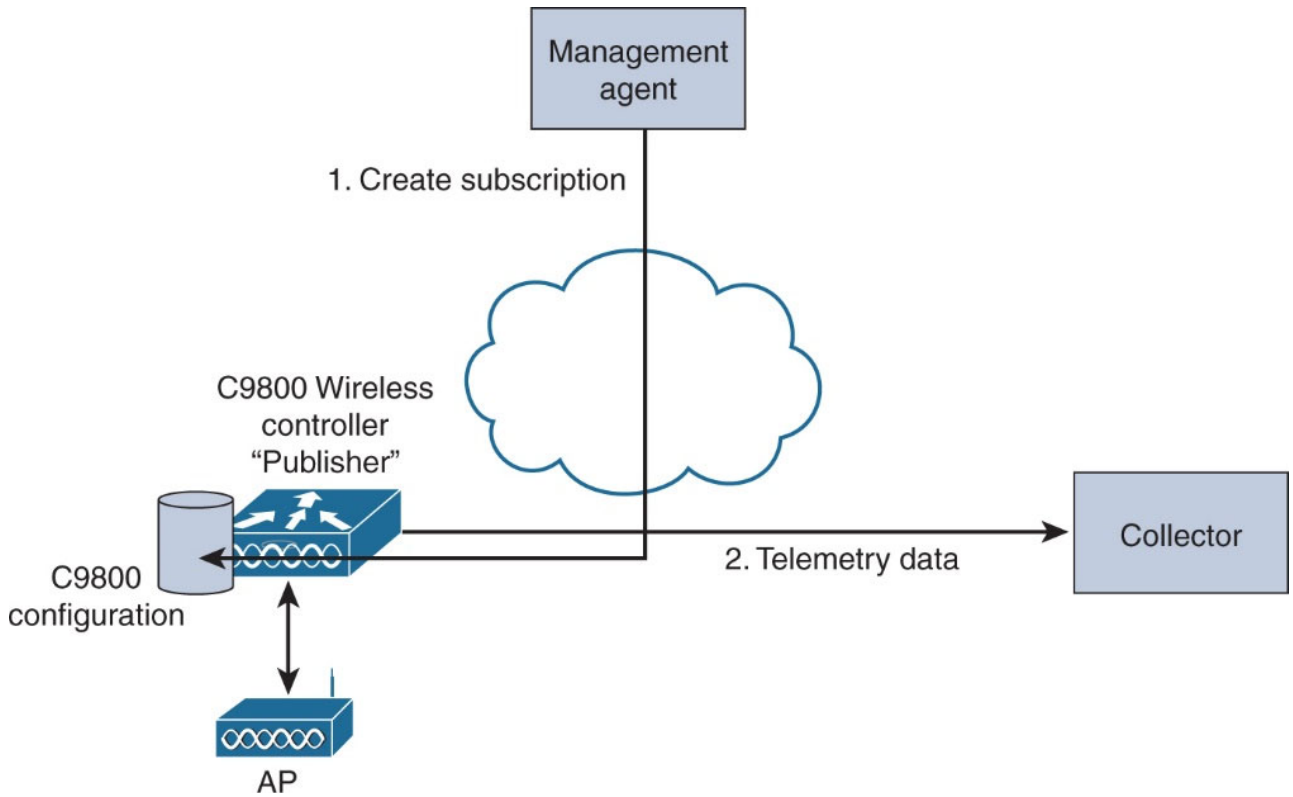


Figure 13-14 Flow for dial-out telemetry subscriptions

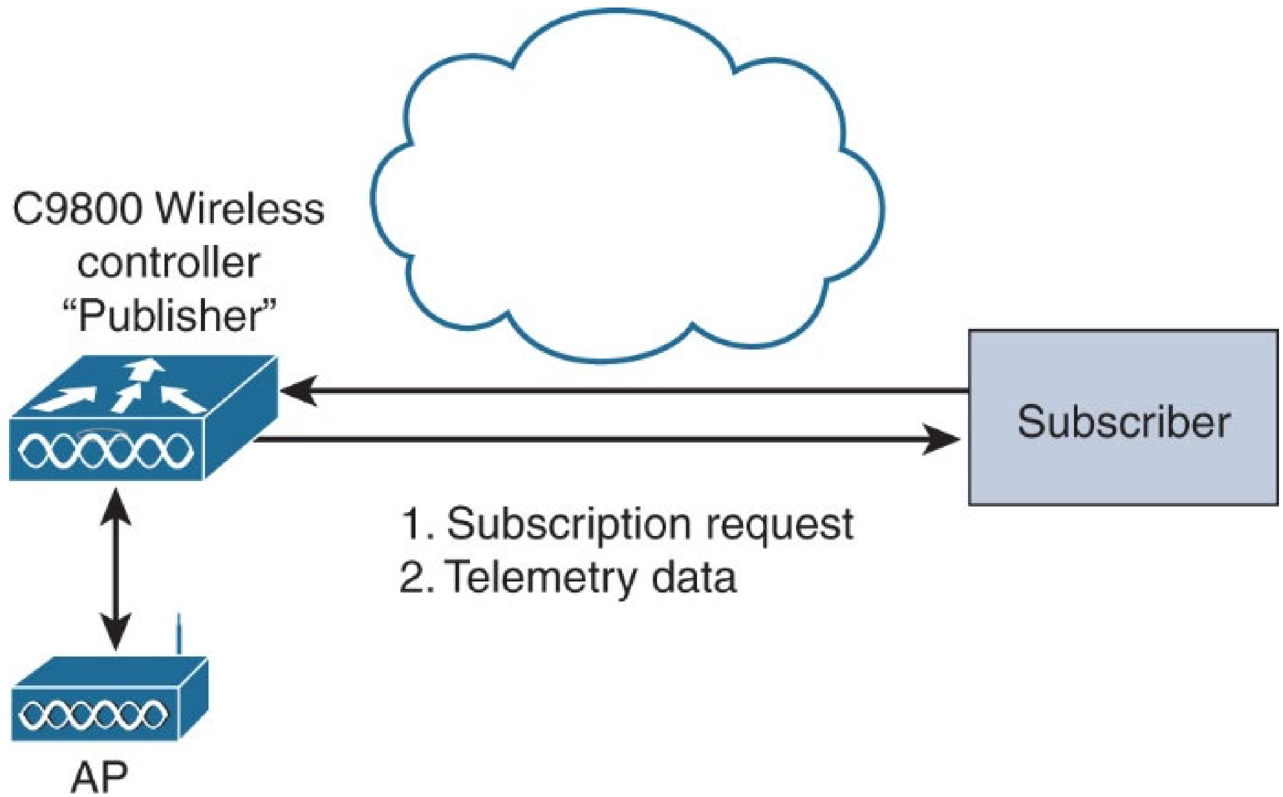


Figure 13-15 Flow for dial-in telemetry subscriptions

sub.py > ...

```python
from ncclient import manager
from ncclient.xml_ import to_ele

rpc = """
<establish-subscription xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
 xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <stream>yp:yang-push</stream>
    <yp:xpath-filter>
        /wireless-ap-global-oper:ap-global-oper-data/ap-join-stats/ap-join-info/ap-name
    </yp:xpath-filter>
    <yp:period>500</yp:period>
</establish-subscription>
"""

host = "192.168.20.150"
username = "lab"
password = "lab"

with manager.connect(host=host,
                        port=830,
                        username=username,
                        password=password,
                        timeout=90,
                        hostkey_verify=False) as m:

    response = m.dispatch(to_ele(rpc))
    print("Waiting for notify...")
    while True:
        n = m.take_notification()
        print(n.notification_xml)

```

Figure 13-16 Python code to create a dial-in telemetry subscription

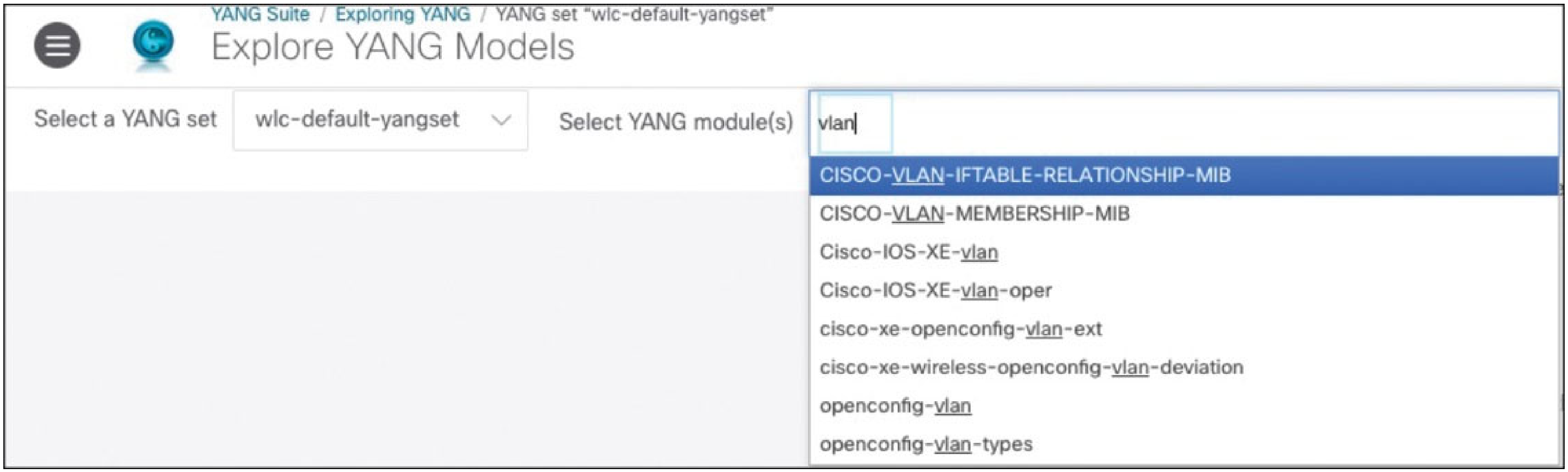


Figure 13-17 Searching models in YANG Suite

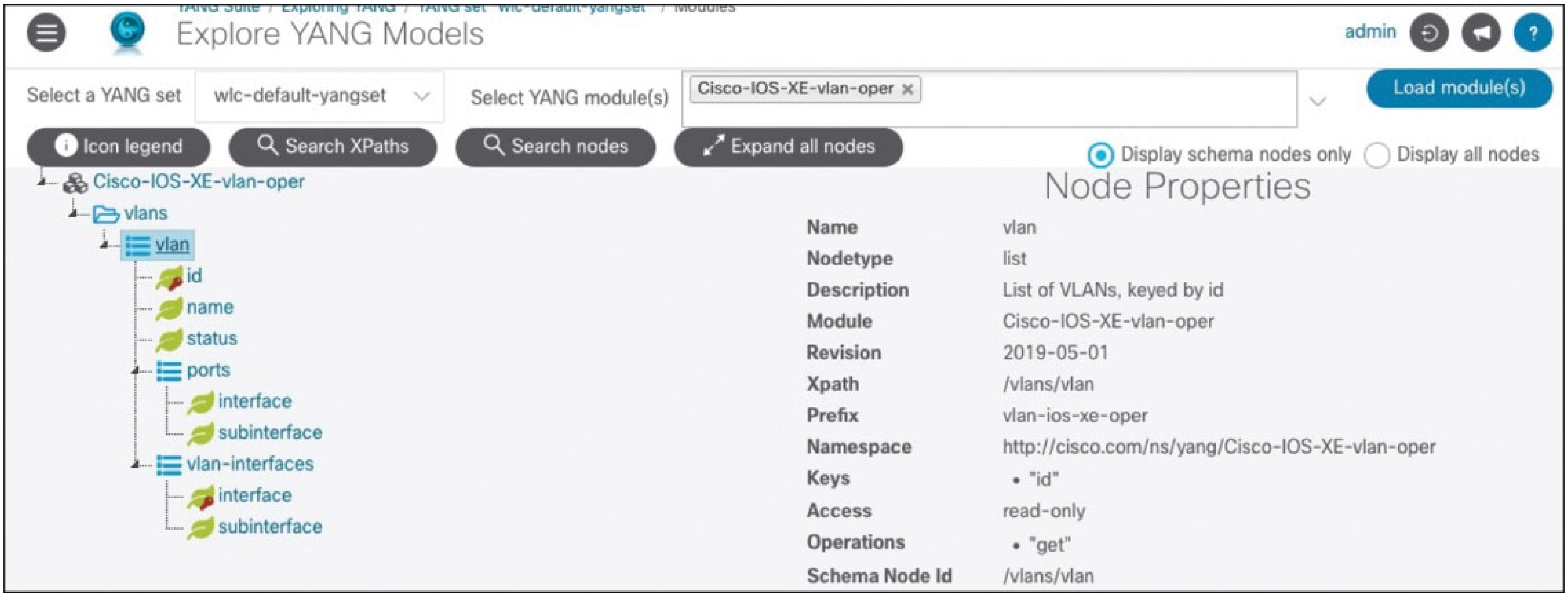


Figure 13-18 Displaying details of YANG models using YANG Suite

```
rpc = """
<establish-subscription xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
 xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <stream>yp:yang-push</stream>
    <yp:xpath-filter>
        /vlan-ios-xe-oper:vlans/vlan
    </yp:xpath-filter>
    <yp:dampening-period>0</yp:dampening-period>
</establish-subscription>
"""
```

Figure 13-19 Python code with the subscription NETCONF RPC

```yaml
version: '3'
services:
    yangsuite:
      image: yangsuite:latest
      build:
        context: ./yangsuite
      env_file:
        - ./yangsuite/setup.env
      command: /yangsuite/migrate_and_start.sh
      ports:
        - "50052:50052"
        - "50051:50051"
        - "9339:9339"
        - "57344:57344"
        - "57345:57345"
        - "443:443"
```

Figure 13-20 Ports exposed in a docker-compose file

From *Understanding and Troubleshooting Cisco Catalyst 9800 Series Wireless Controller by* Simone Arena, Francisco Sedano Crippa, Nicolas Darchis and Sudha Katgeri(9780137492329) Cisco Press

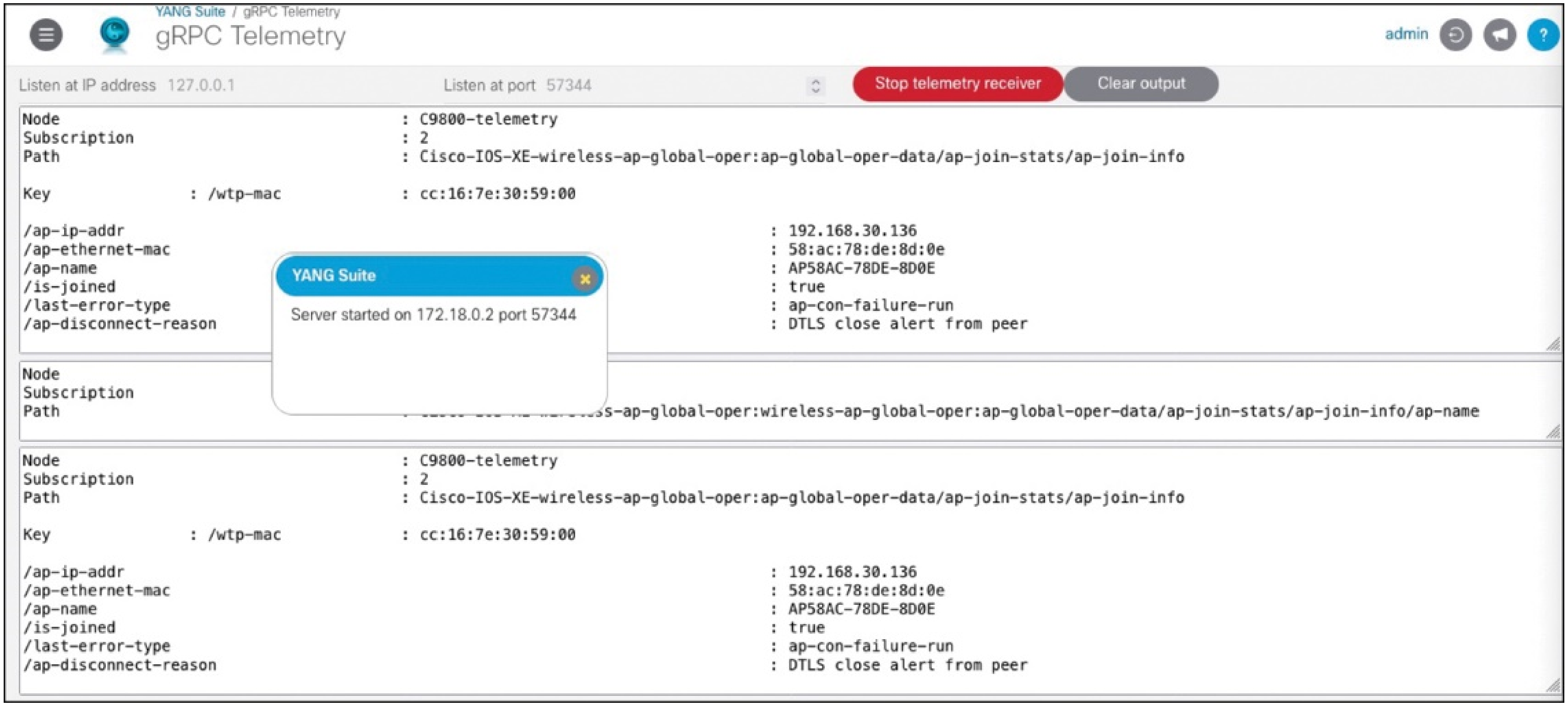


Figure 13-21 YANG Suite receiving gRPC dial-out telemetry

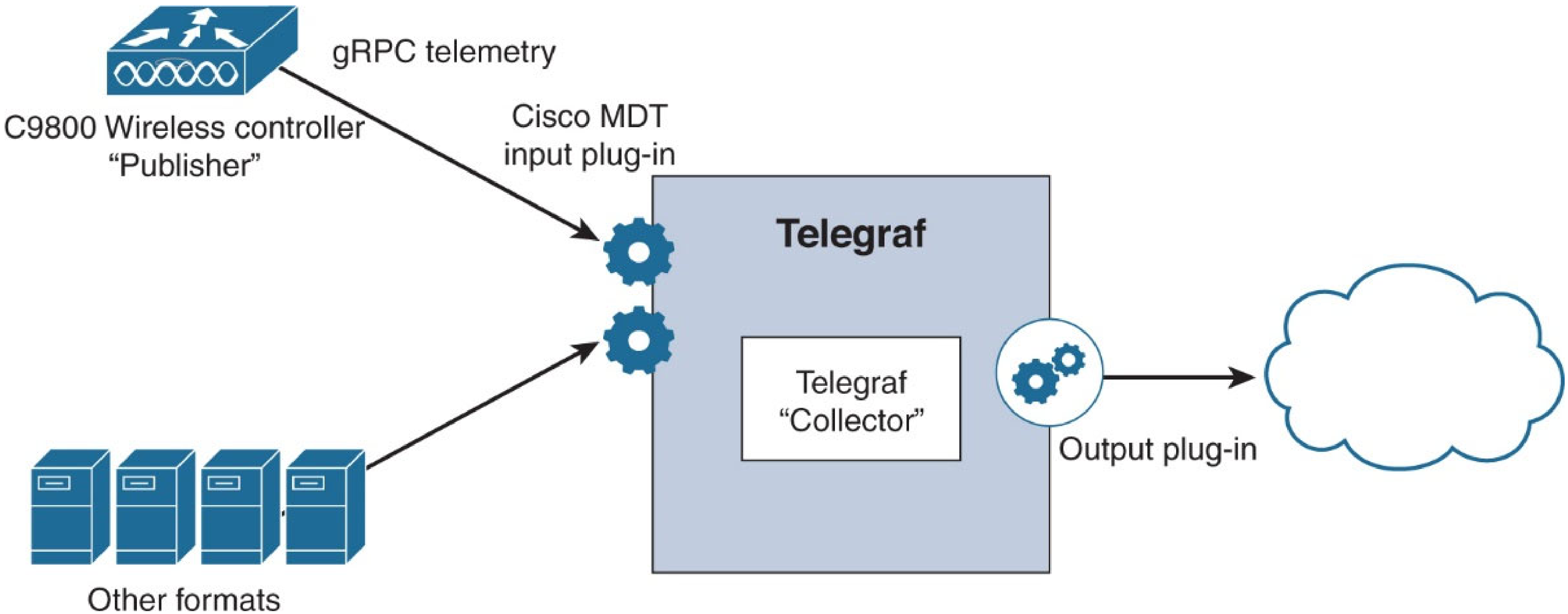


Figure 13-22 Telegraf serving as an aggregator for telemetry data

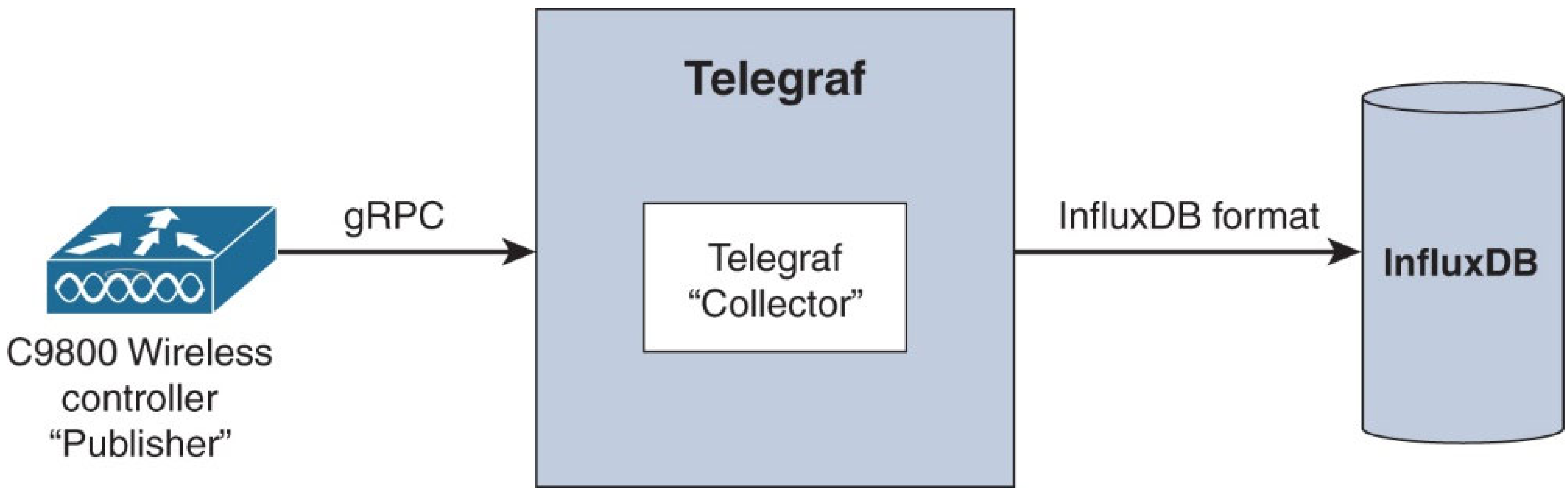


Figure 13-23 Sending data to InfluxDB

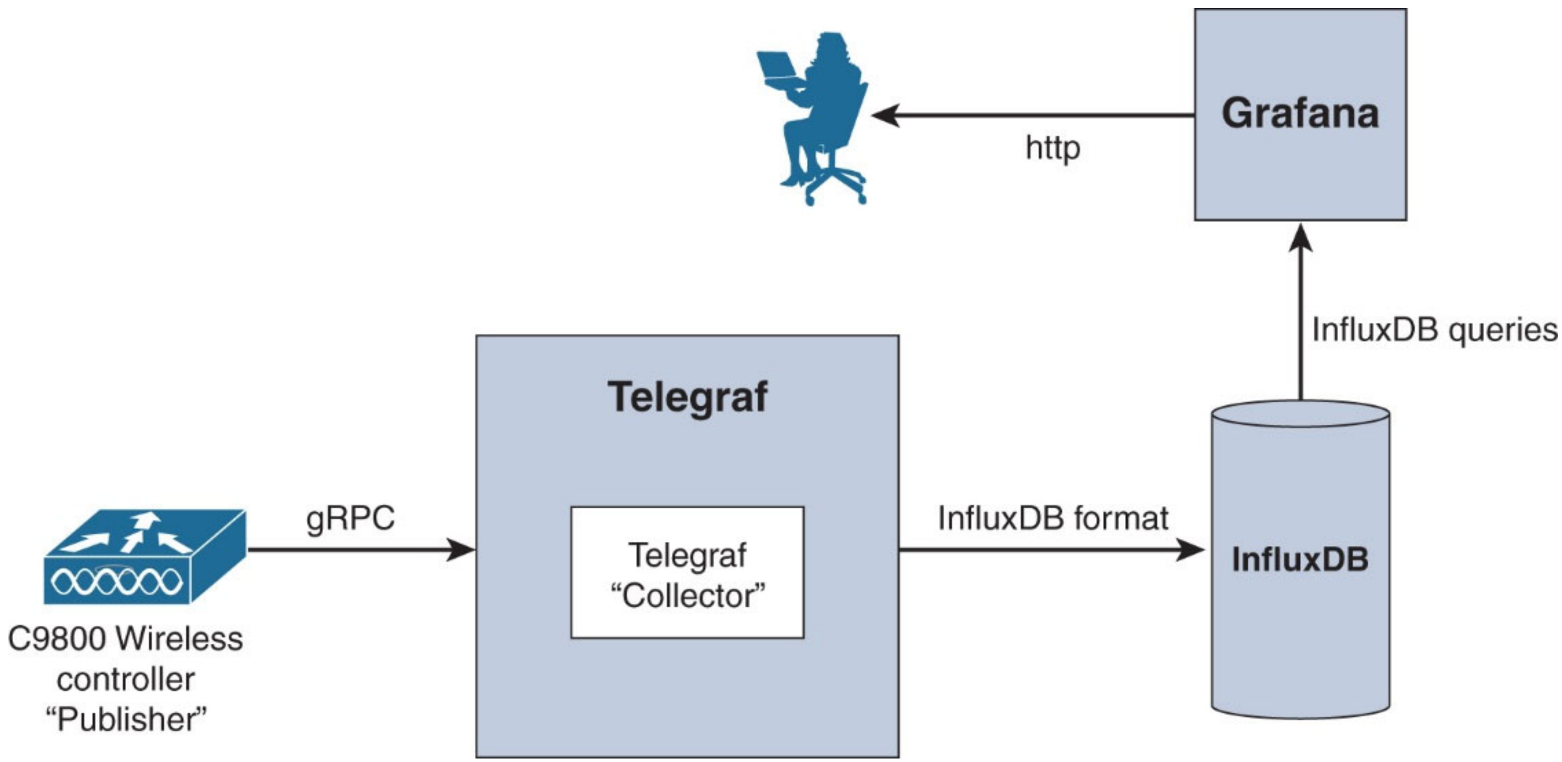


Figure 13-24 Grafana receiving data from InfluxDB

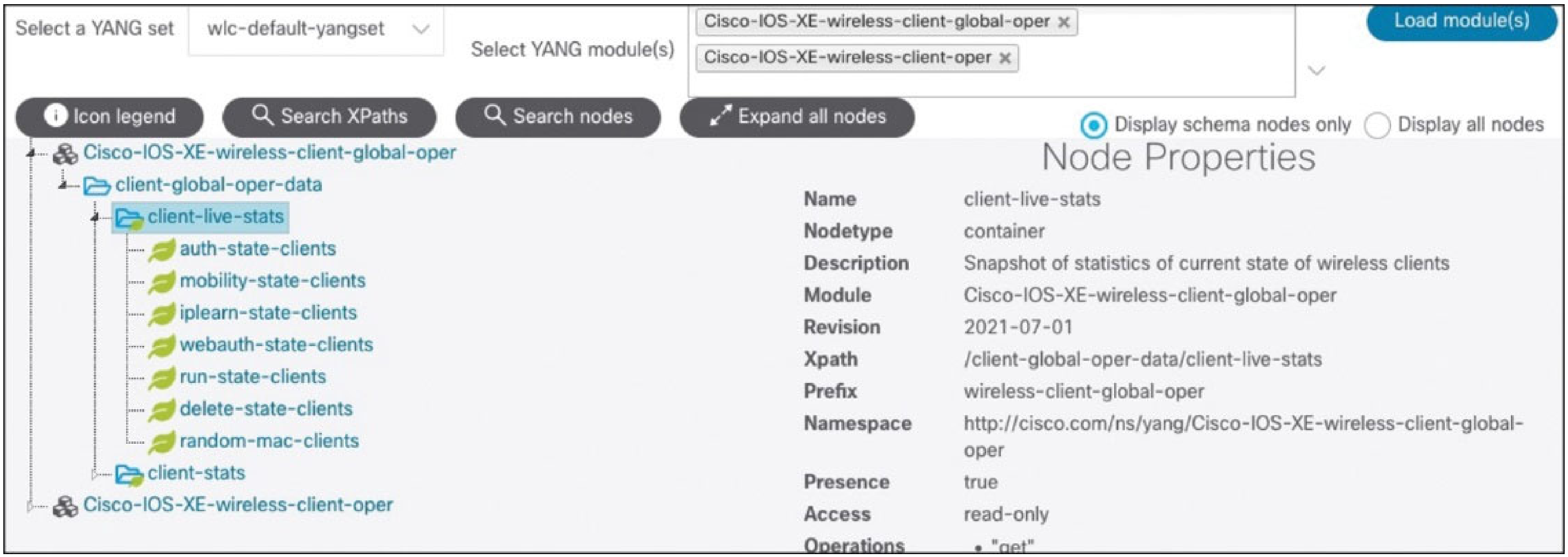


Figure 13-25 Identifying prefix and XPath using YANG Suite

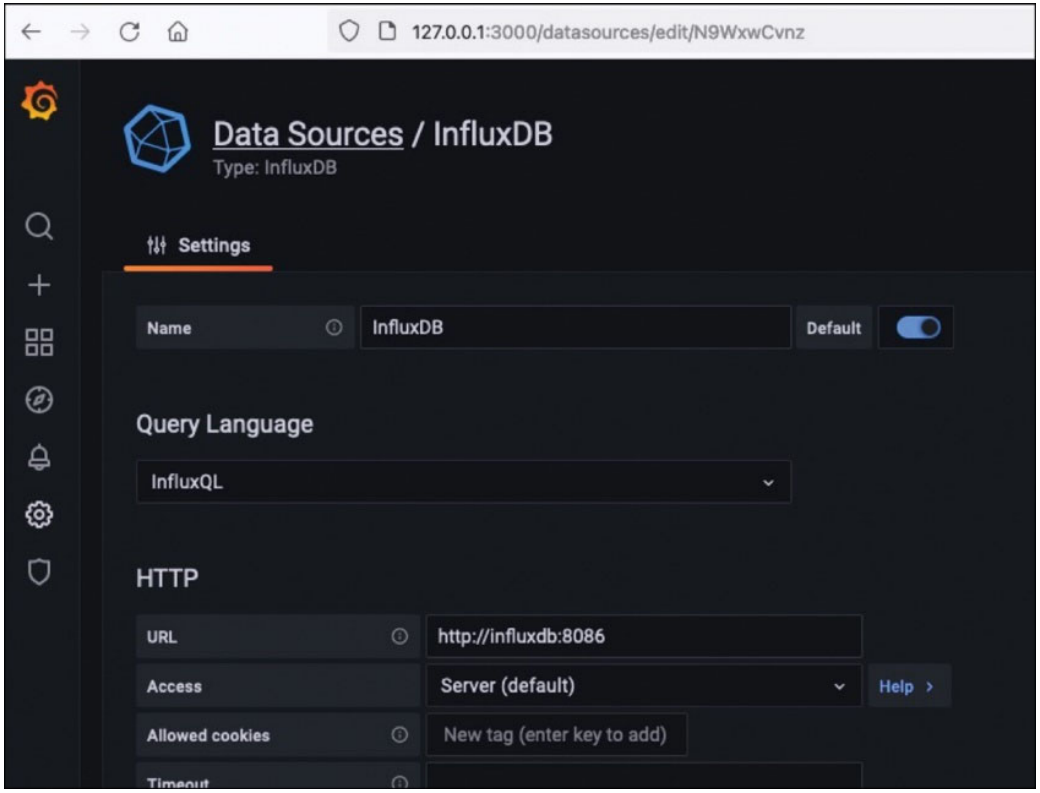


Figure 13-26 Configuring the database connection in Grafana

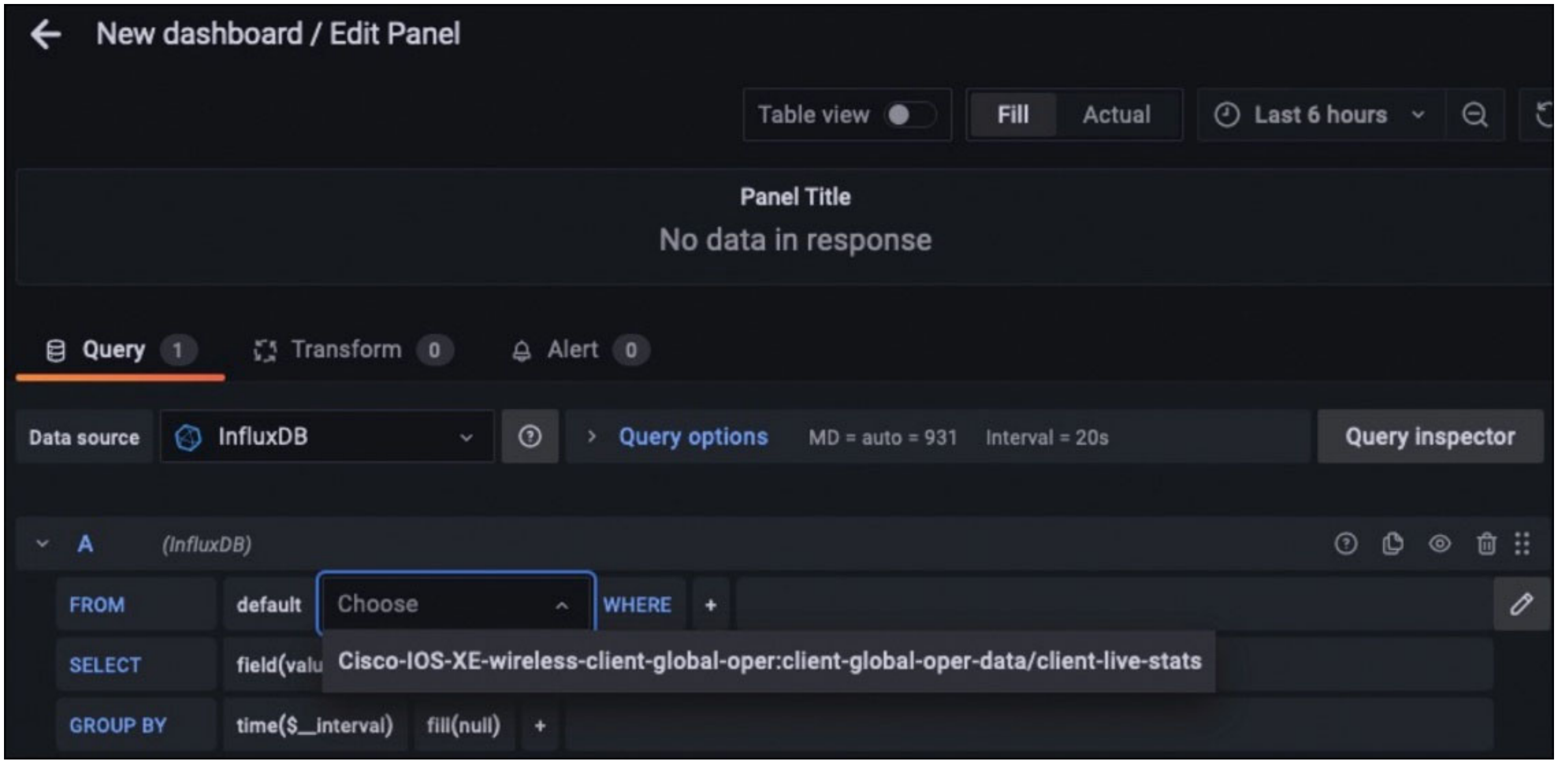


Figure 13-27 Displaying available measurements in Grafana

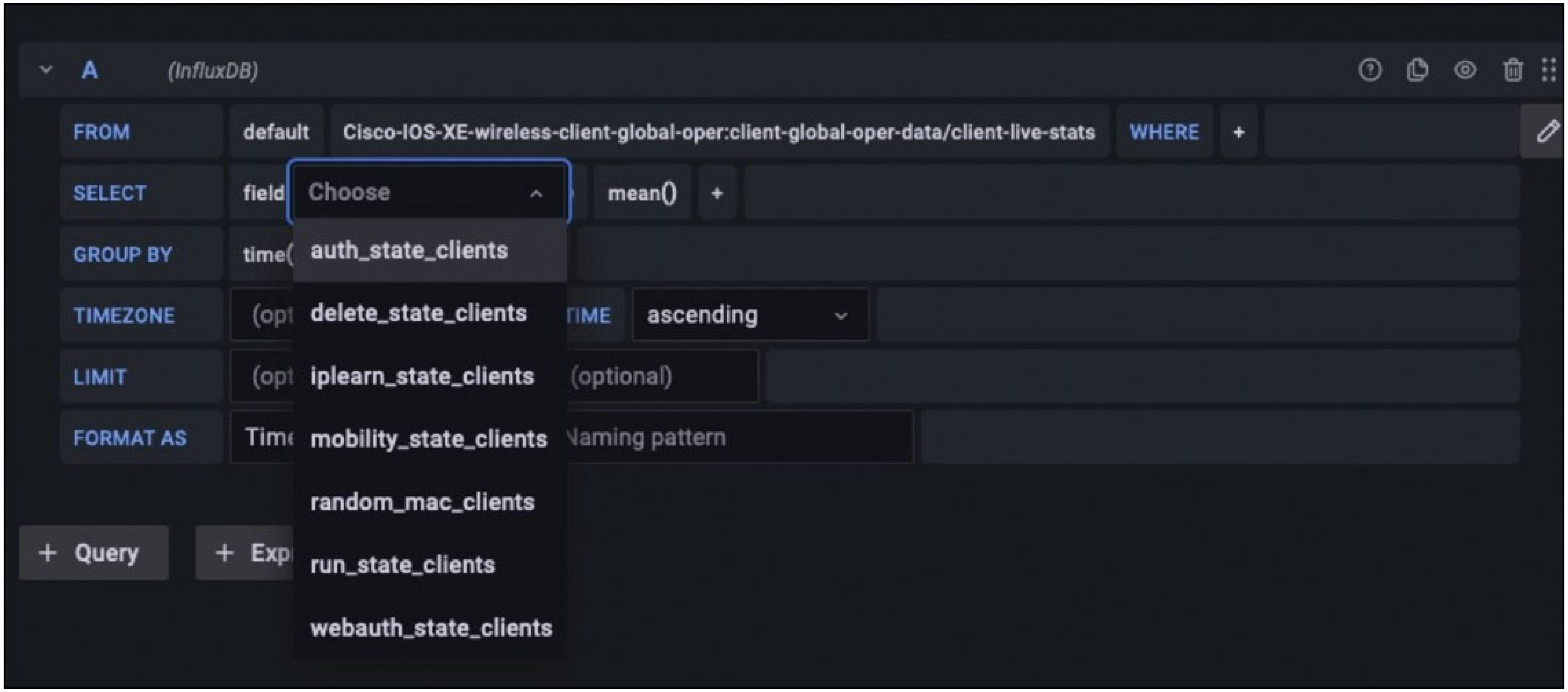


Figure 13-28 Displaying available metrics in Grafana

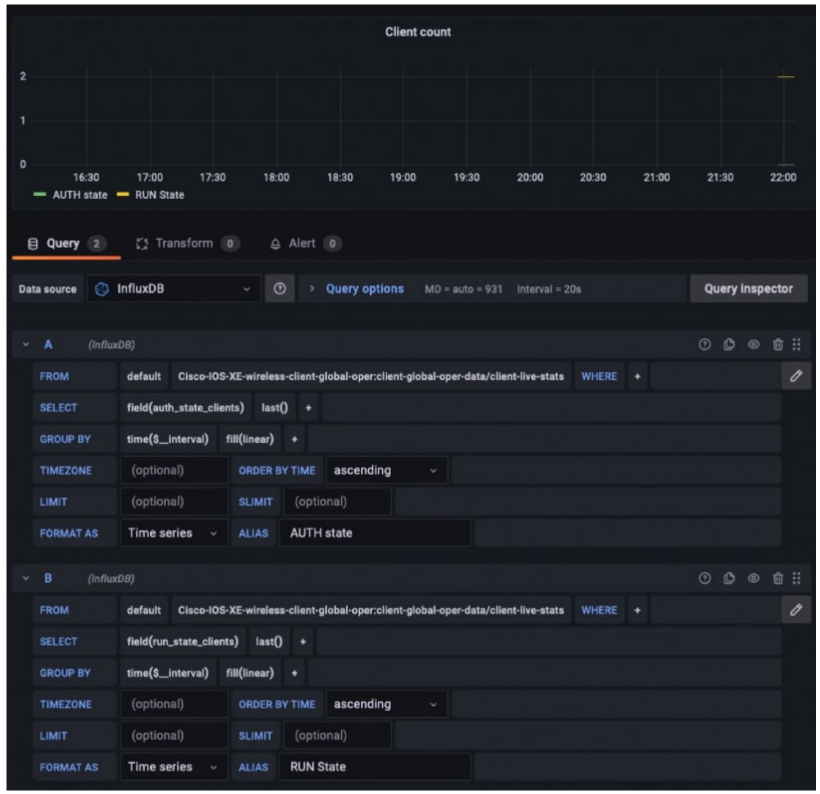


Figure 13-29 Grafana configuration

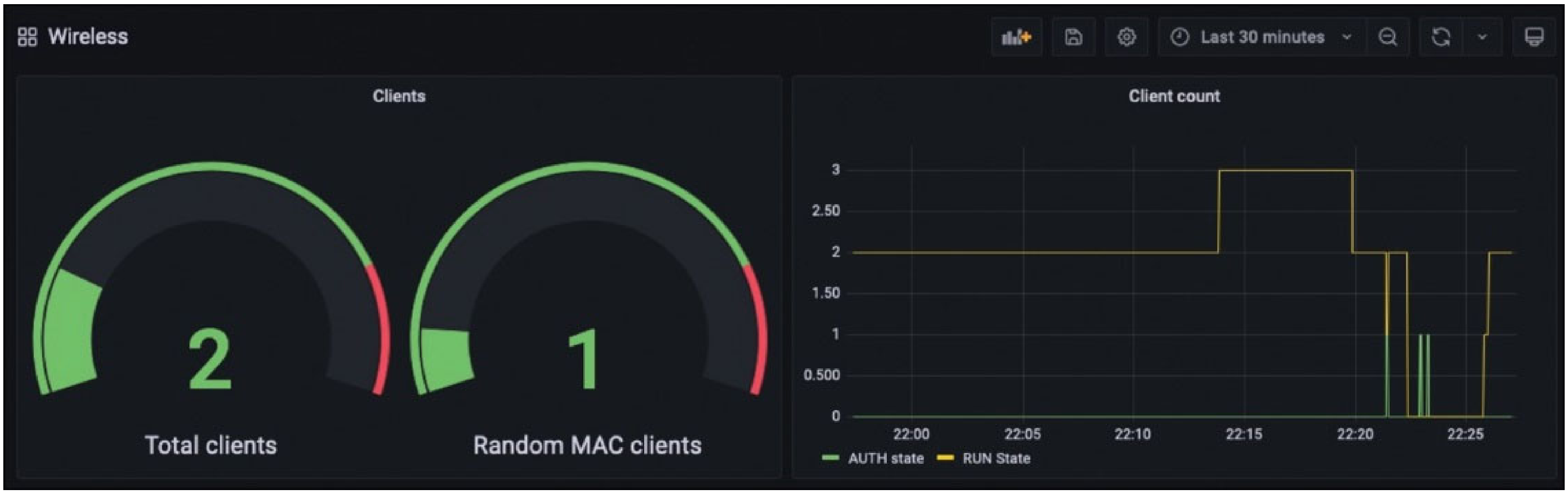


Figure 13-30 Final Grafana dashboard

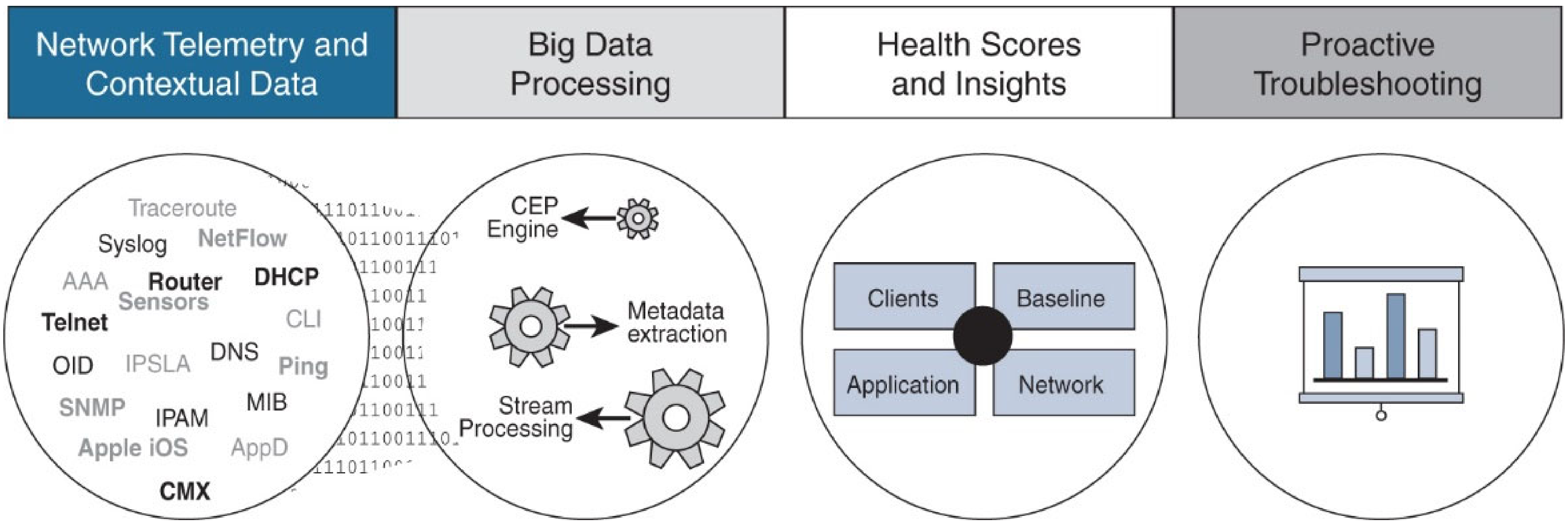


Figure 14-1 The principles of Cisco DNA Center telemetry and data processing flows

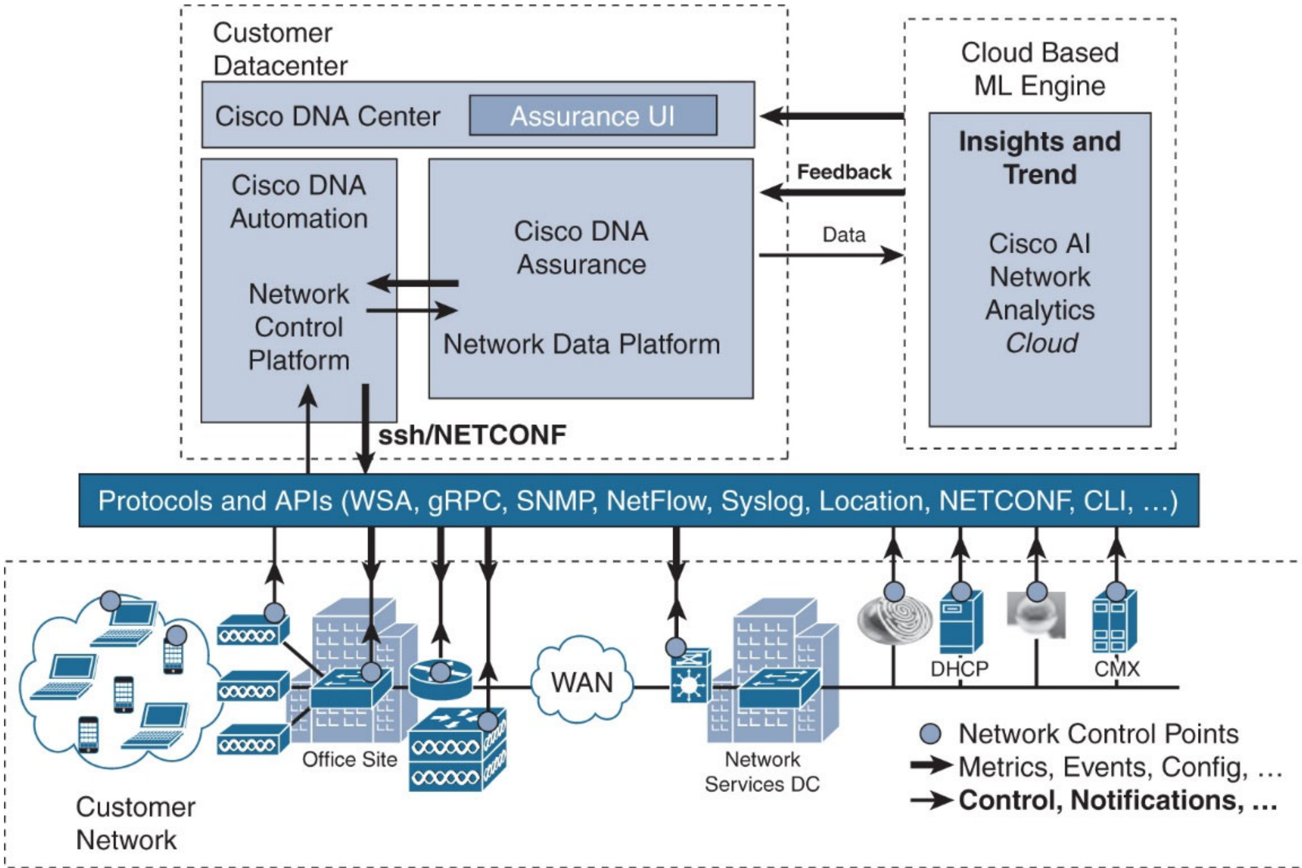


Figure 14-2 Cisco DNA Center Assurance architecture

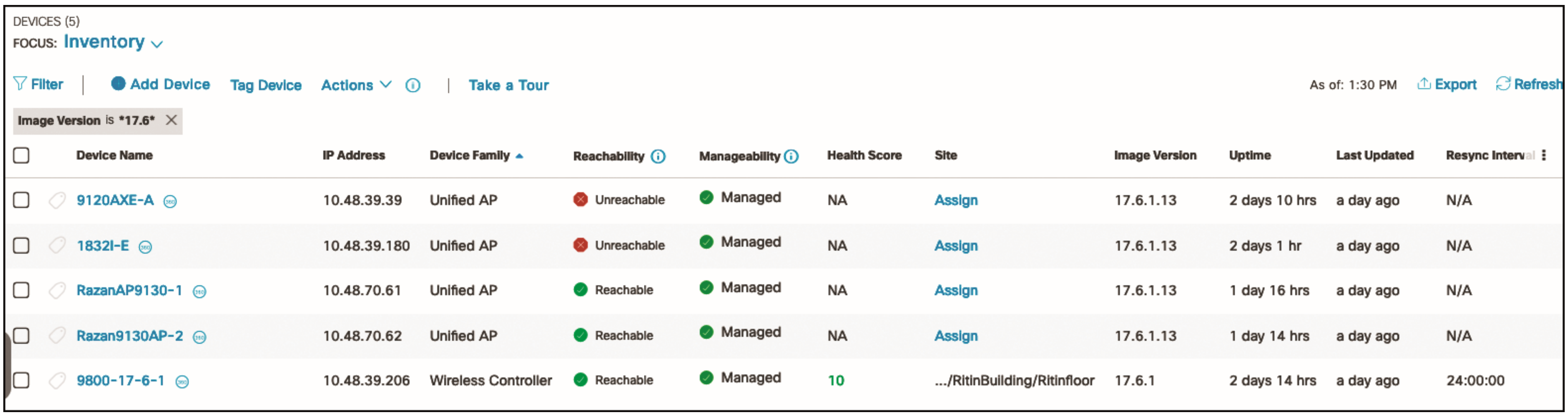


Figure 14-3 Cisco DNA Center inventory page

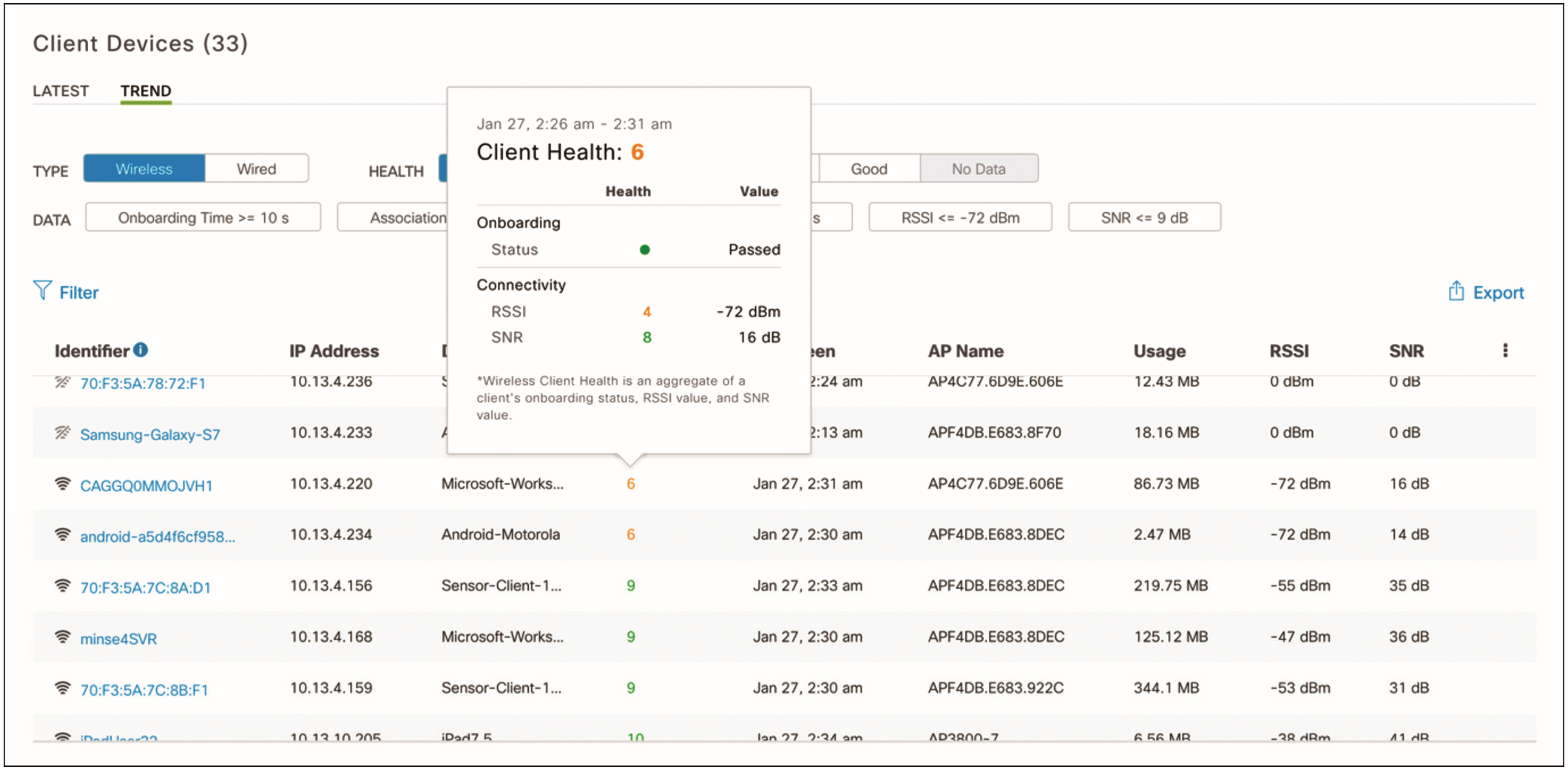


Figure 14-4 Cisco DNA Center client devices list Assurance page

Figure 14-5 Roaming and onboarding times

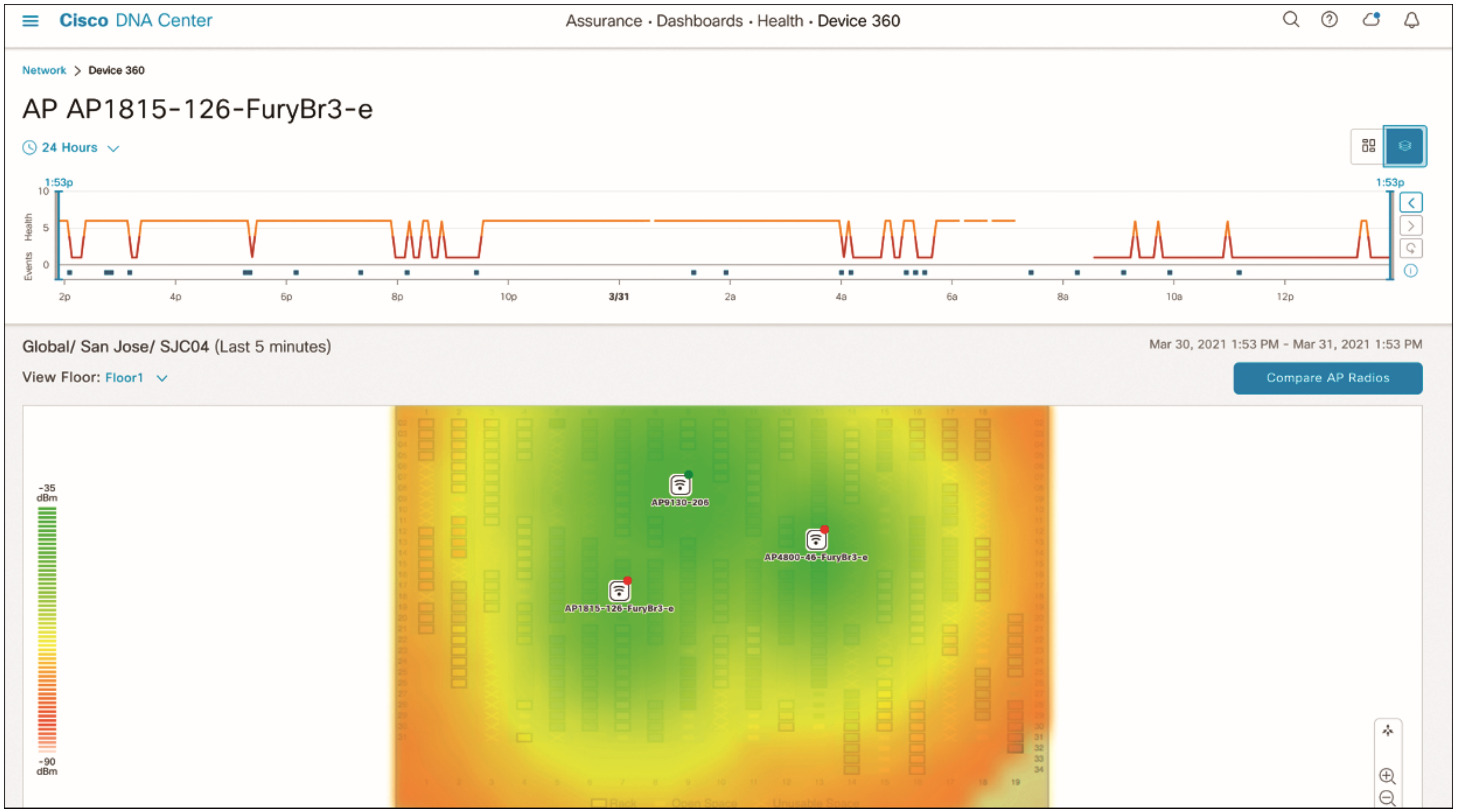


Figure 14-6 AP 360 page

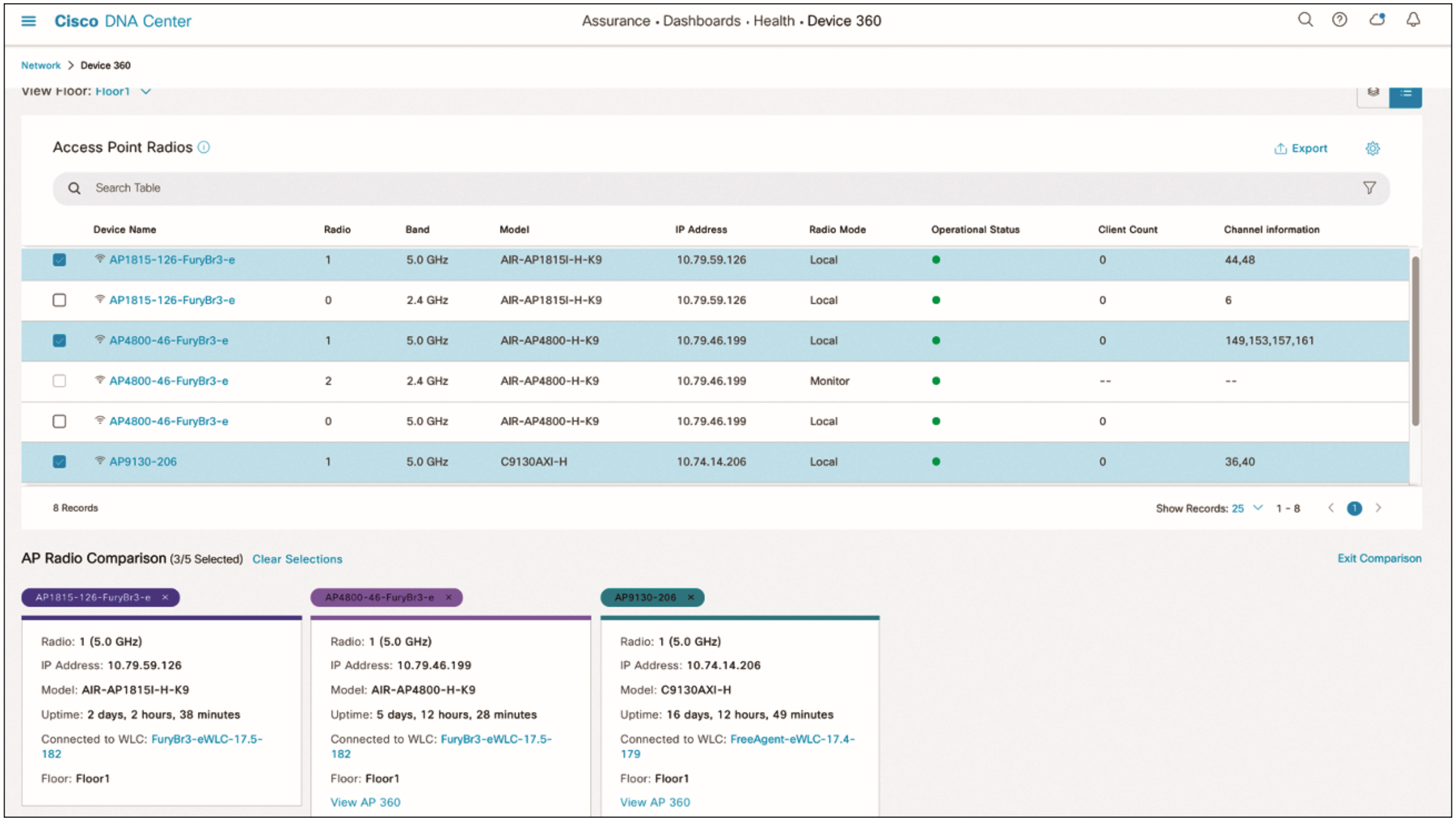


Figure 14-7 AP radios comparison screen

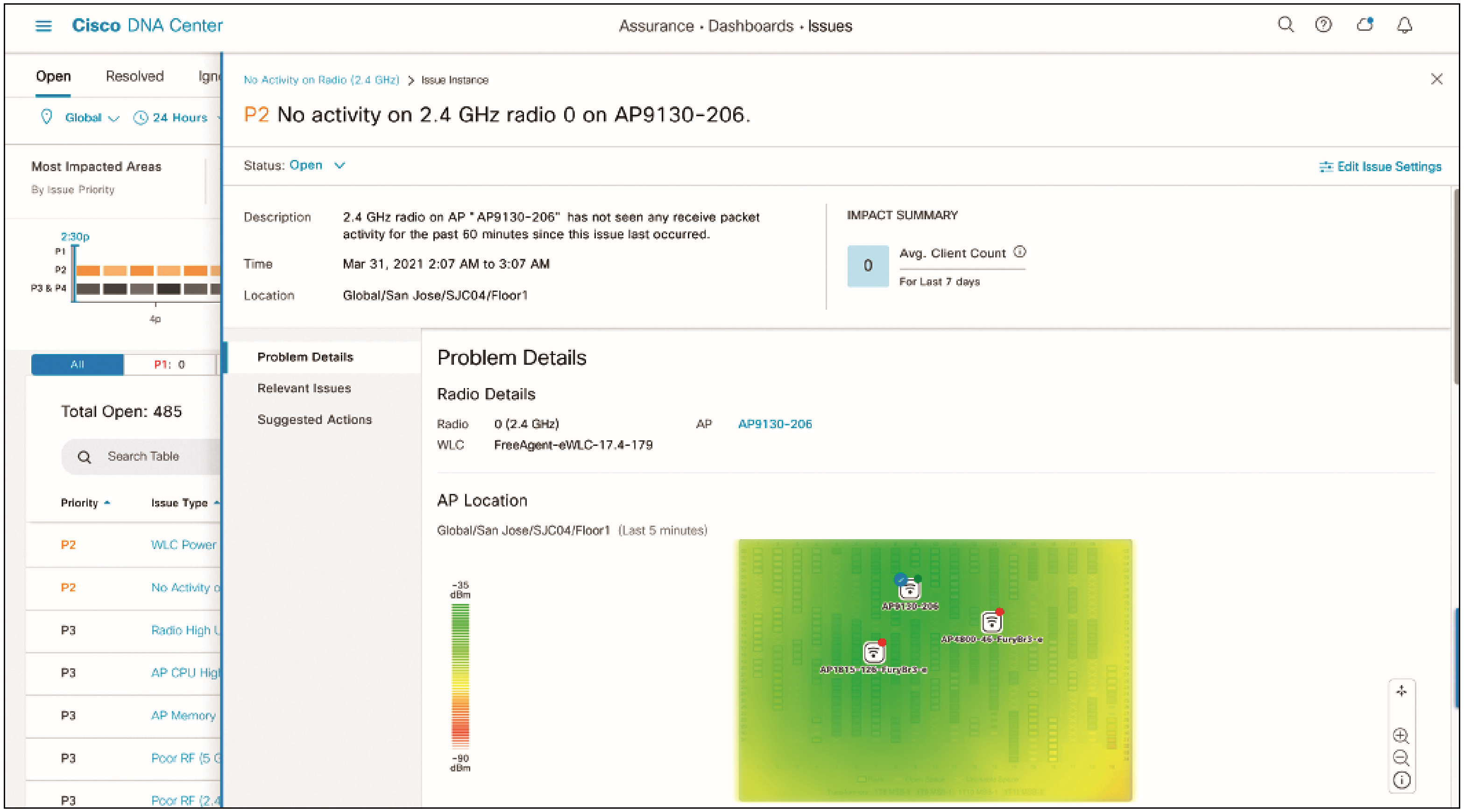


Figure 14-8 Issues dashboard

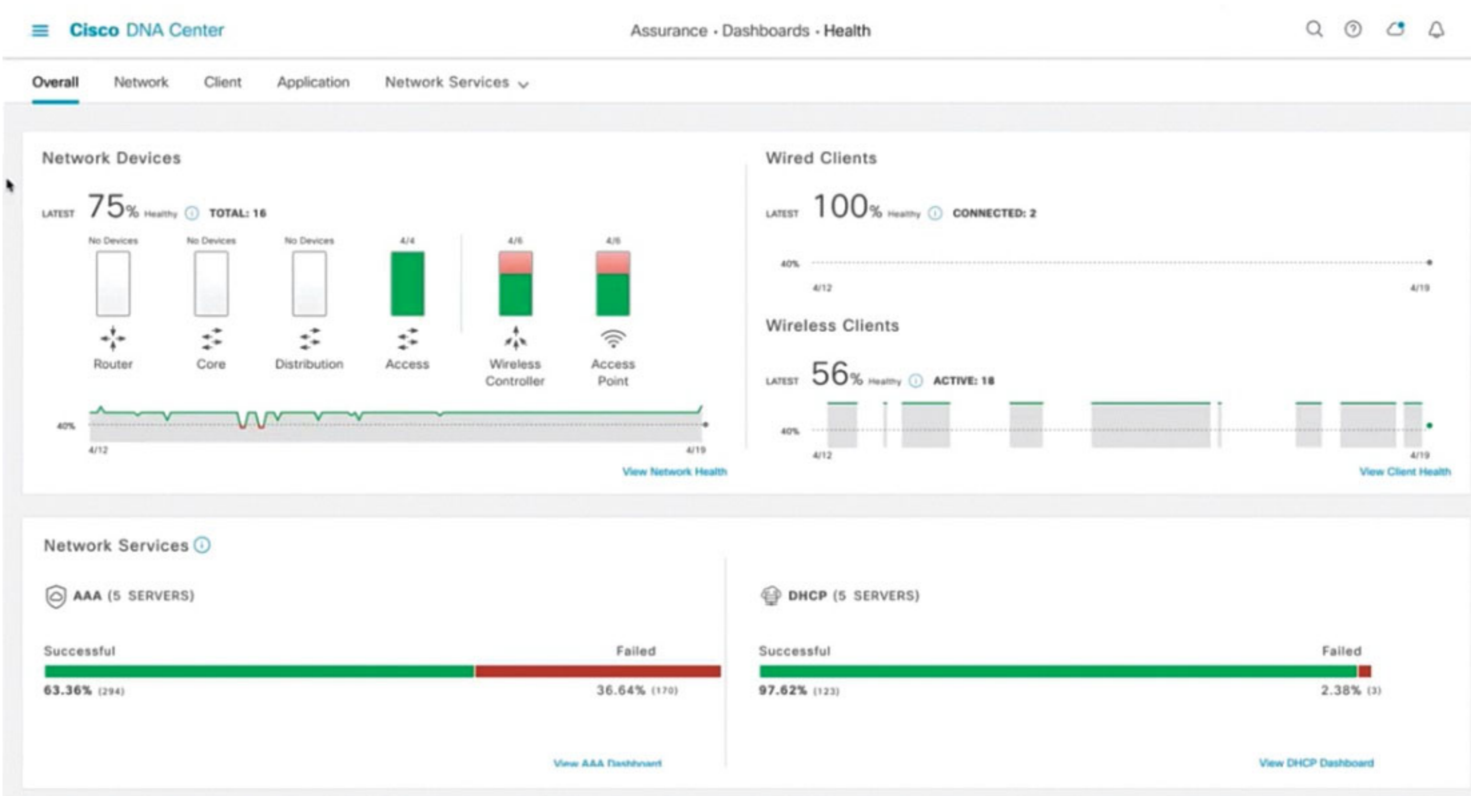


Figure 14-9 Assurance overview health dashboard

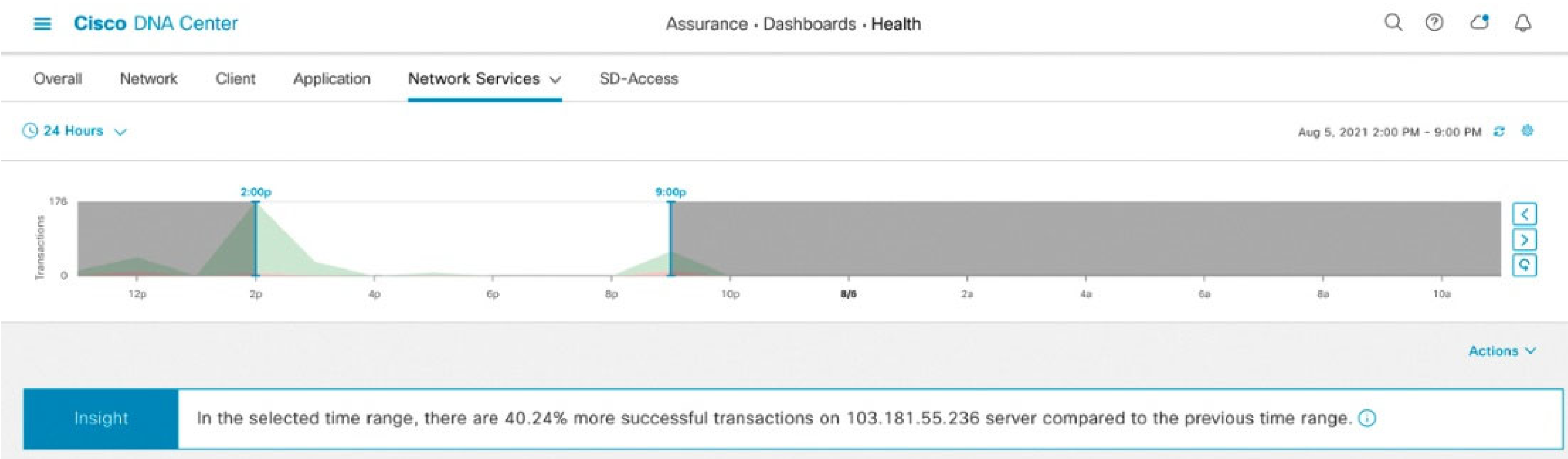


Figure 14-10 Network services health timeline

Figure 14-11 Top AAA sites sorted

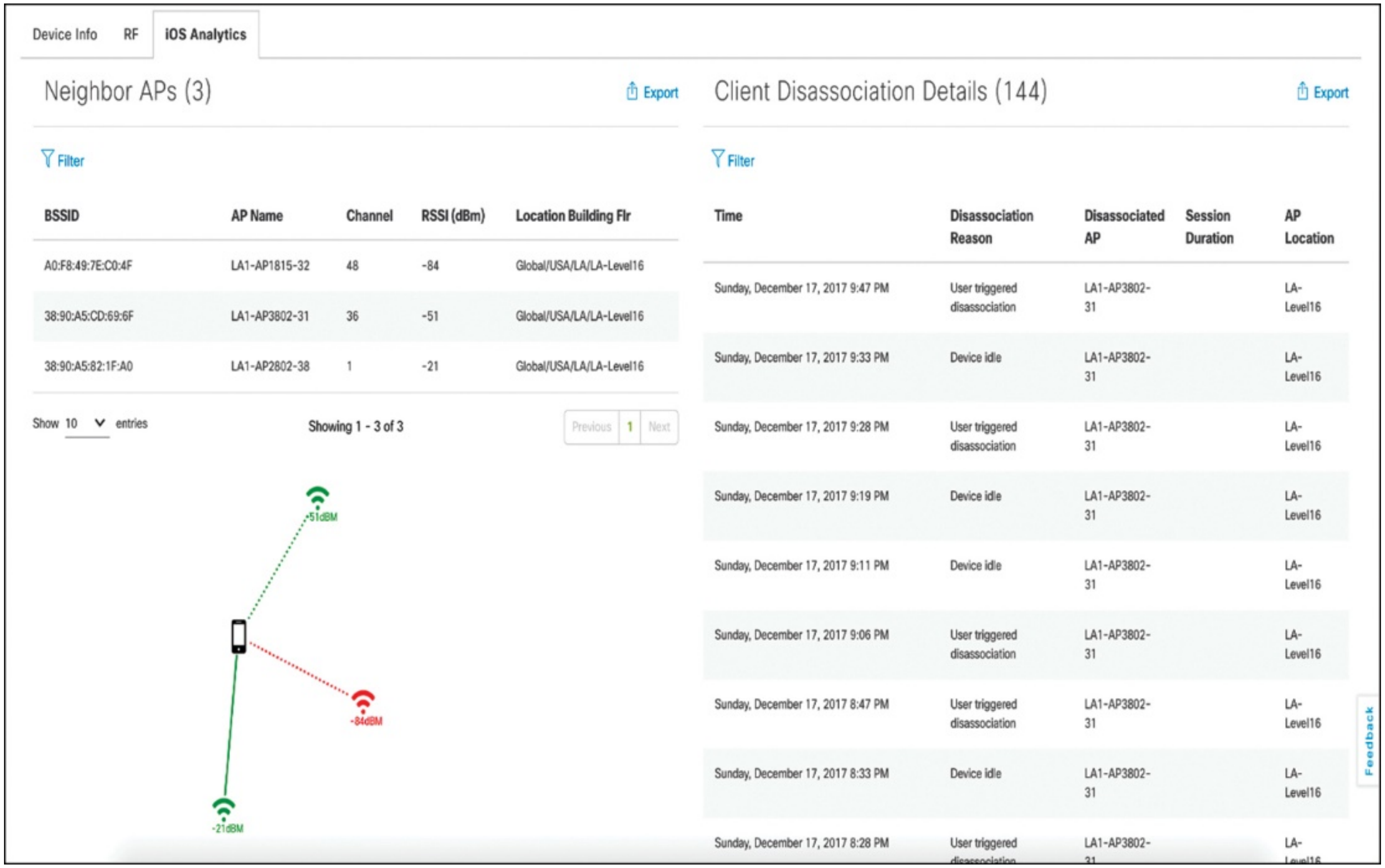


Figure 14-12 The IOS Analytics tab showing the insights received from an iPhone device

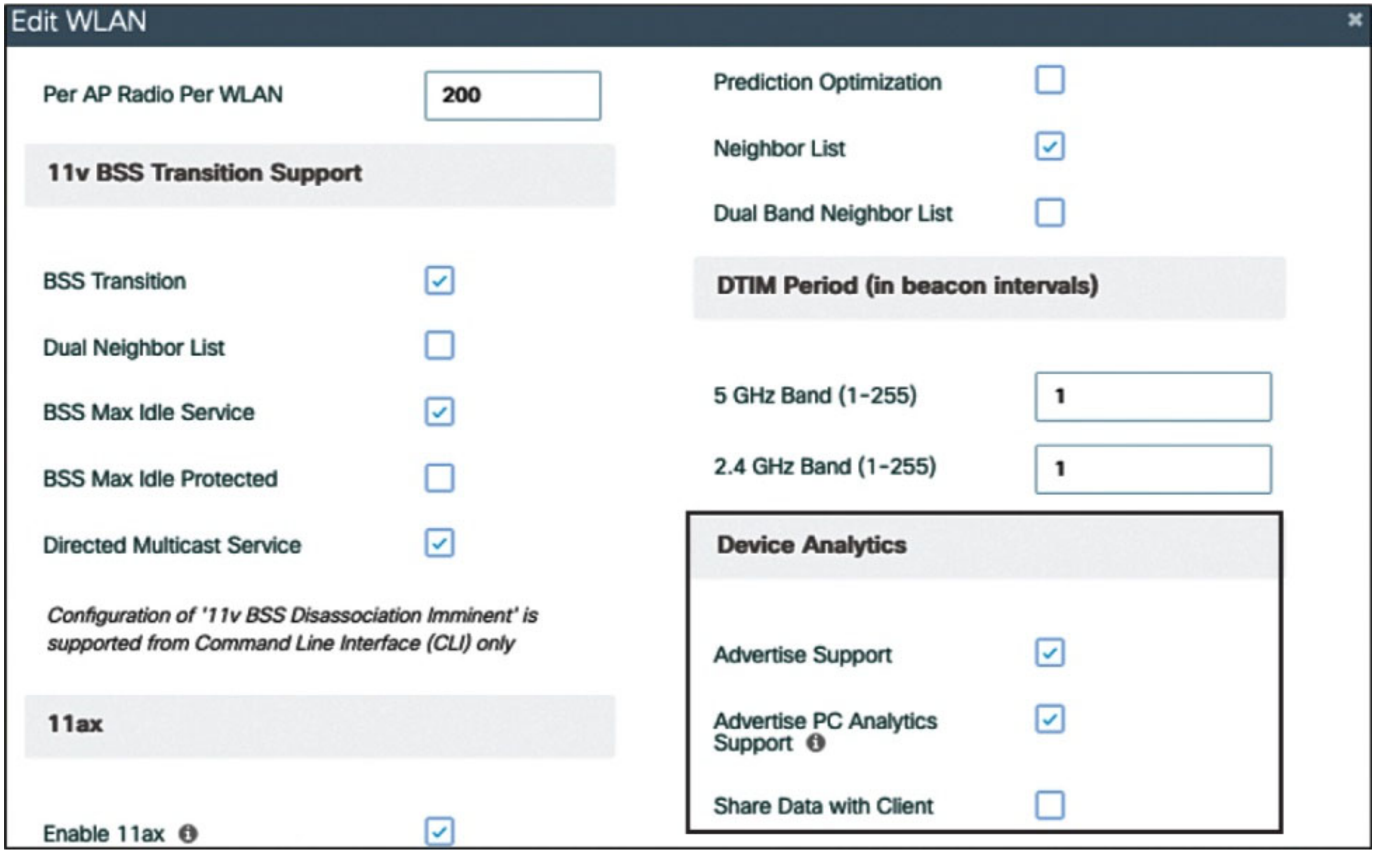


Figure 14-13 Advanced WLAN settings page, allowing you to enable Device Analytics on the C9800

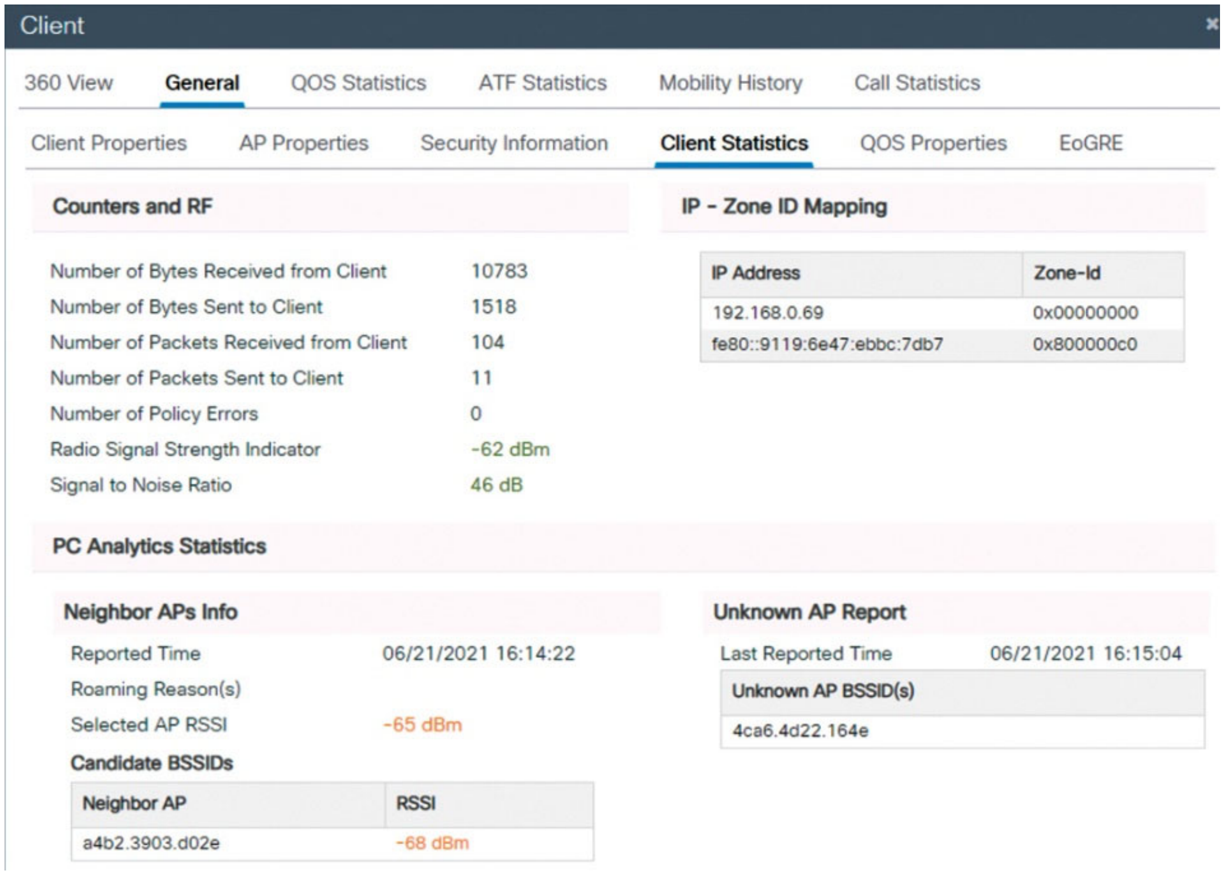


Figure 14-14 C9800 Client statistics monitoring page

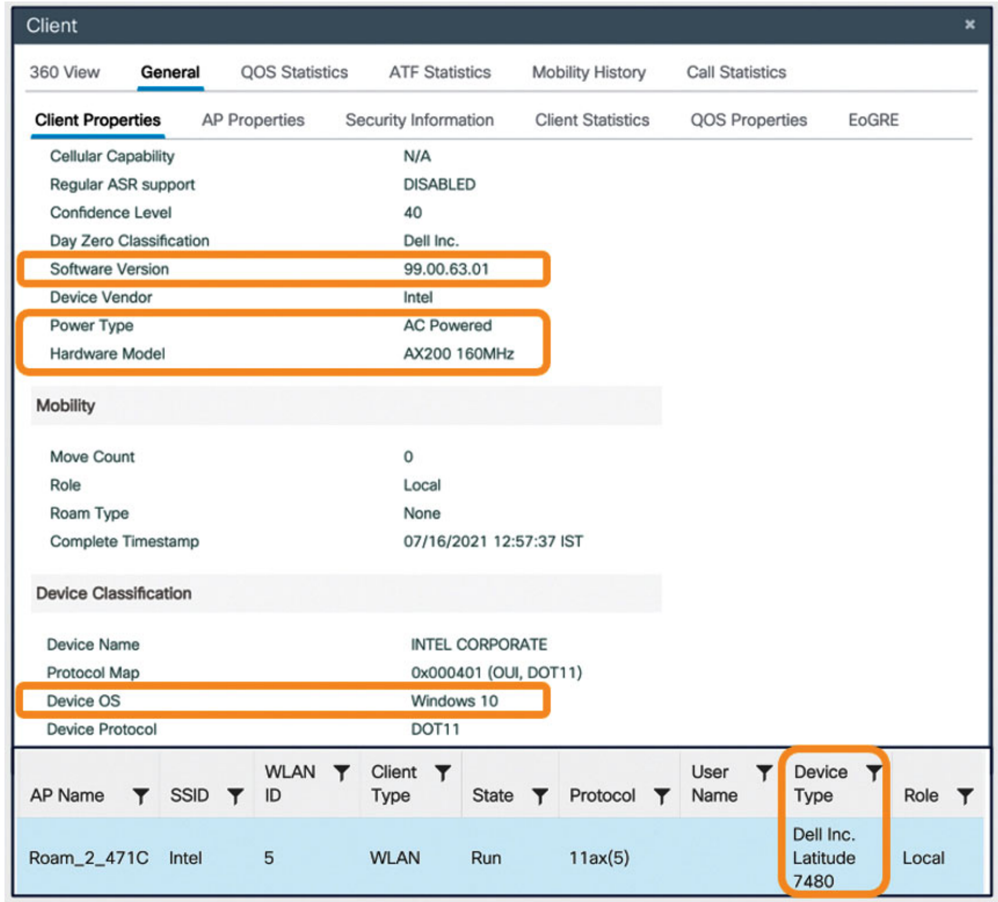


Figure 14-15 C9800 Client general properties

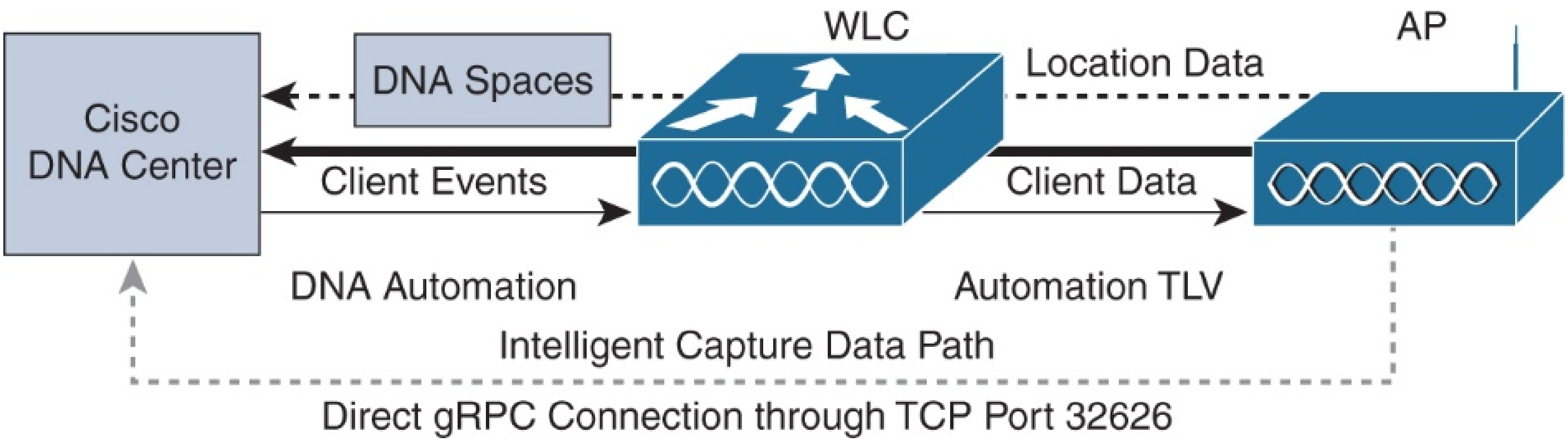


Figure 14-16 Intelligent Capture workflow

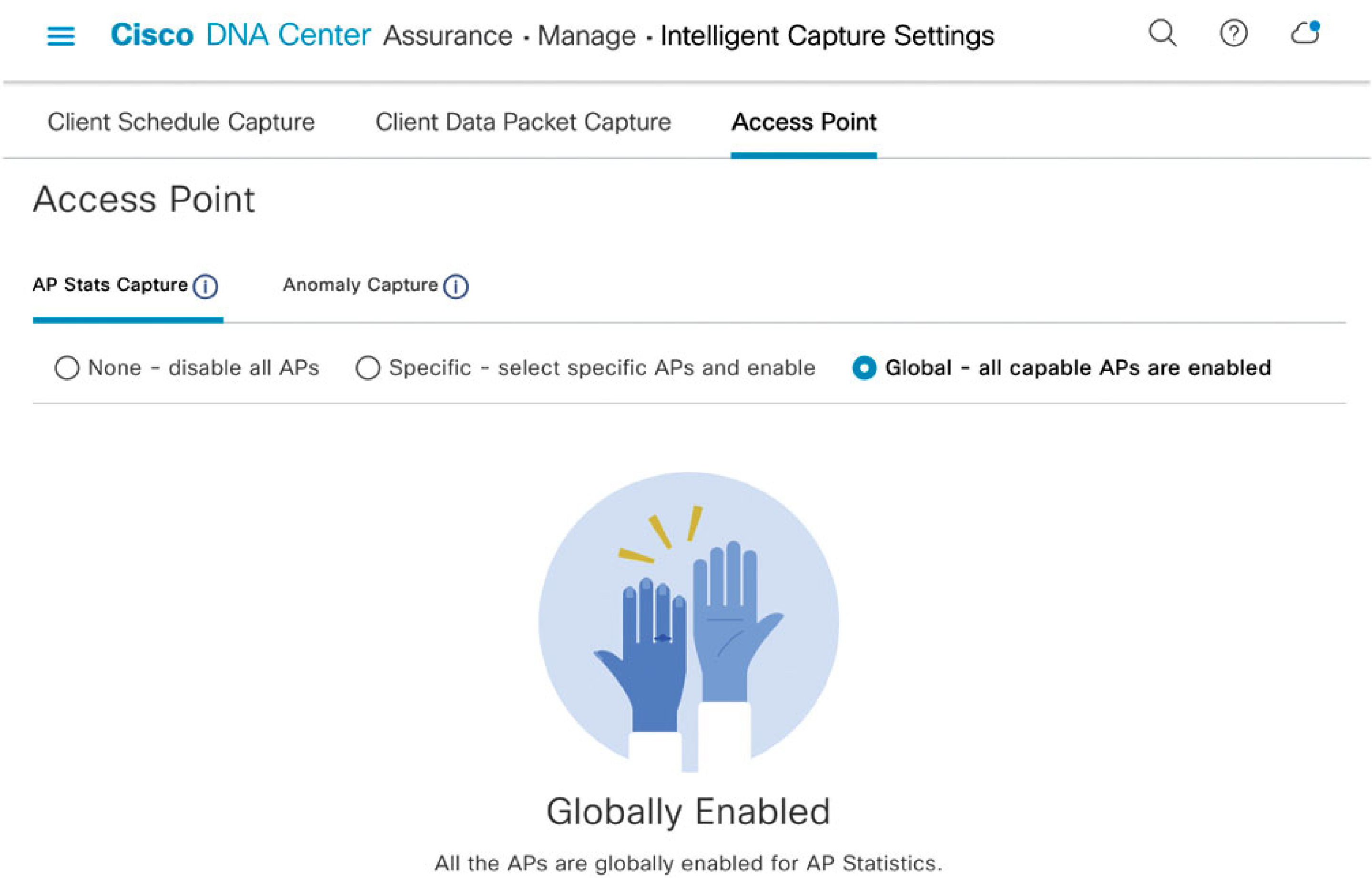


Figure 14-17 Intelligent Capture access point configuration page

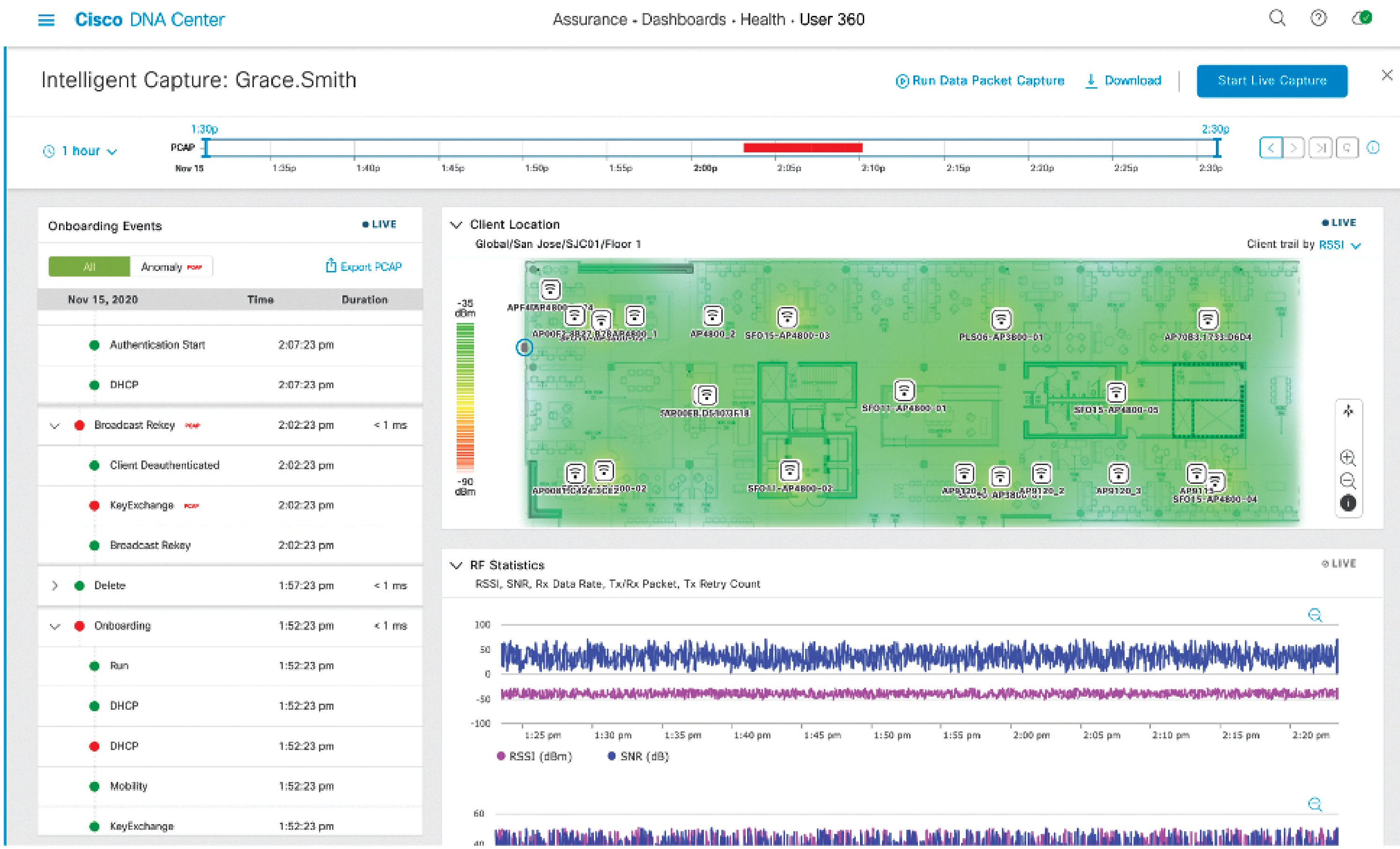


Figure 14-18 Intelligent Capture client 360 health page

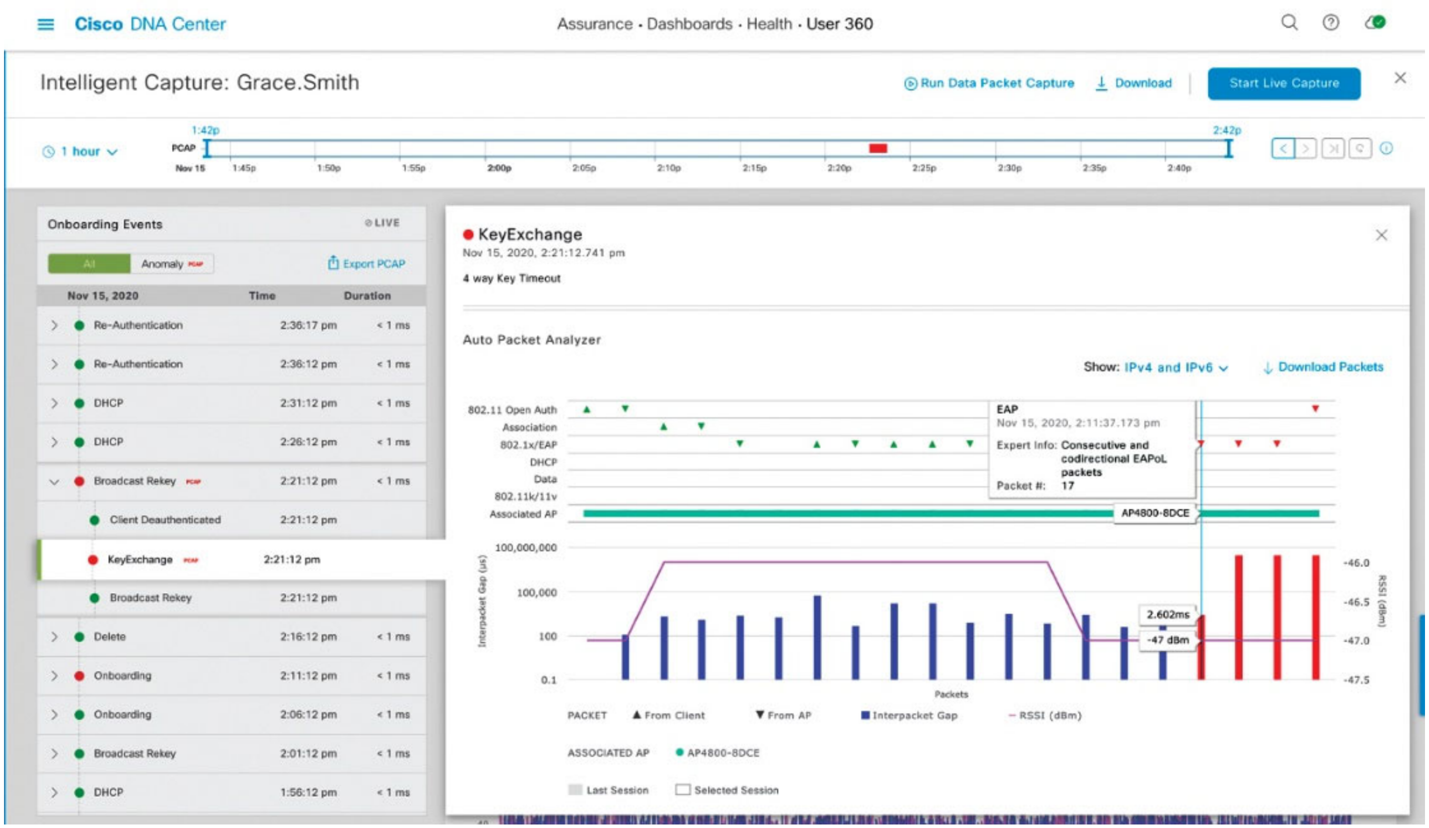


Figure 14-19 The packet analyzer from the onboarding events list

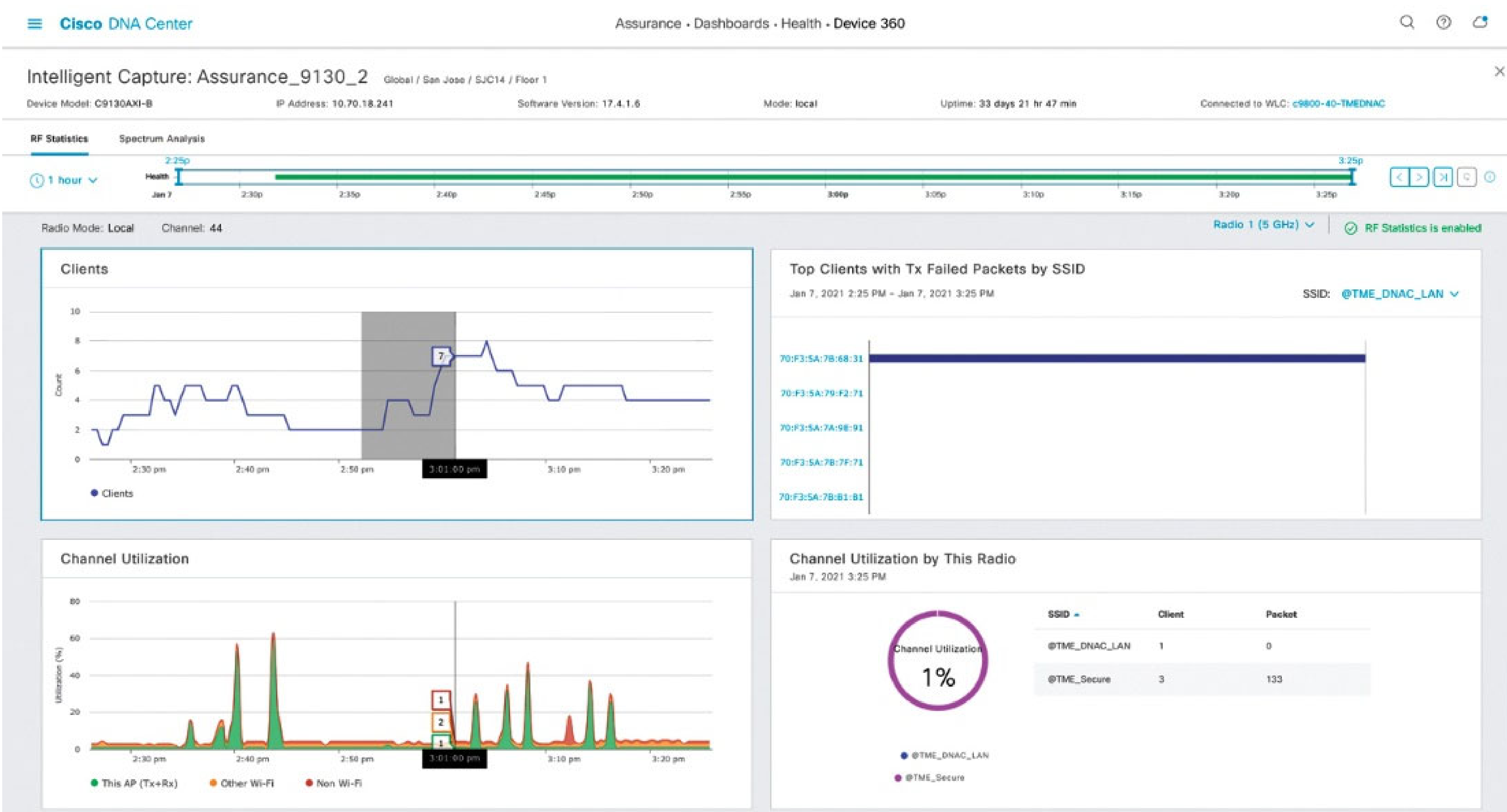


Figure 14-20 The AP health Intelligent Capture page

Figure 14-21 The 1800S sensor

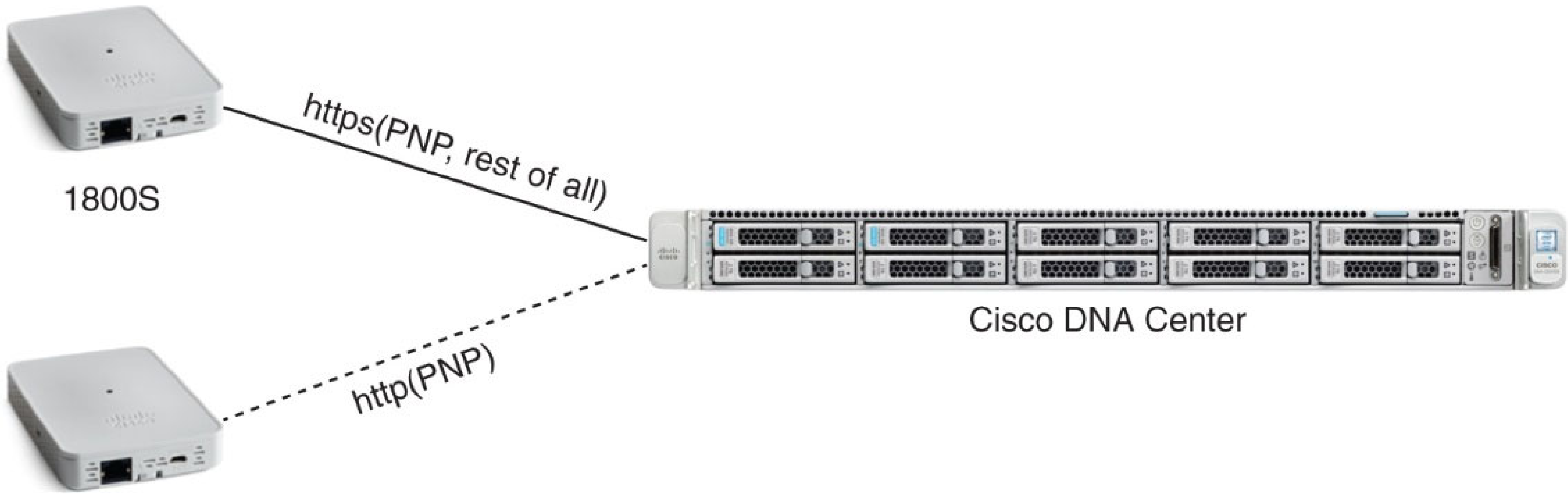


Figure 14-22 1800S protocol workflow overview

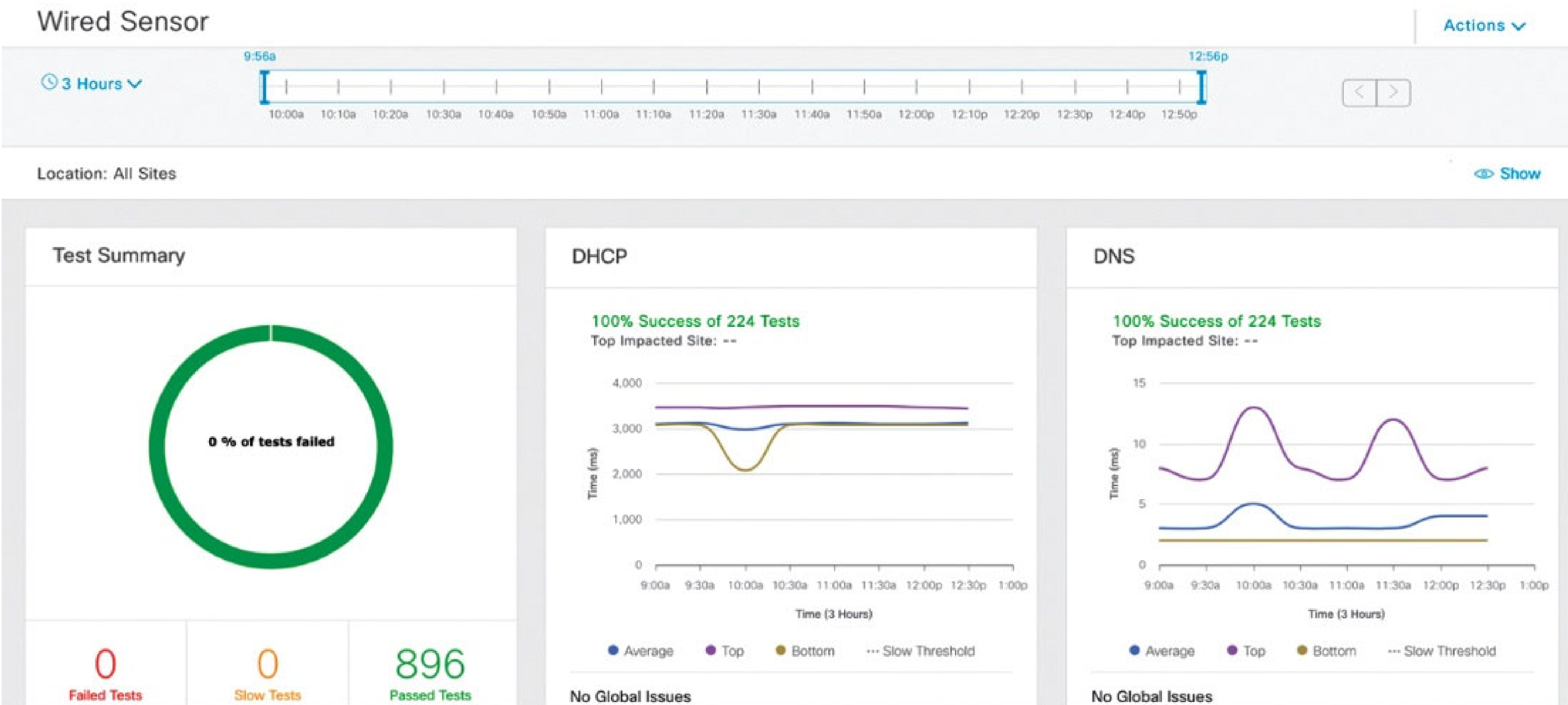


Figure 14-23 Cisco 1800S test suite dashboard

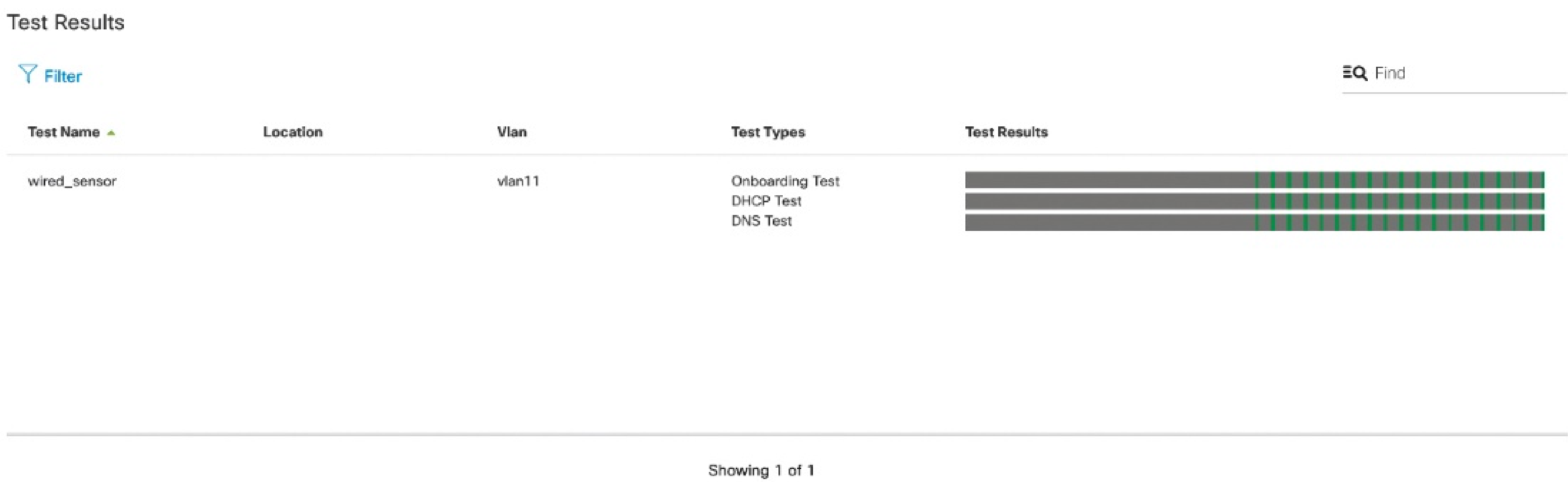


Figure 14-24 Cisco 1800S test result timeline

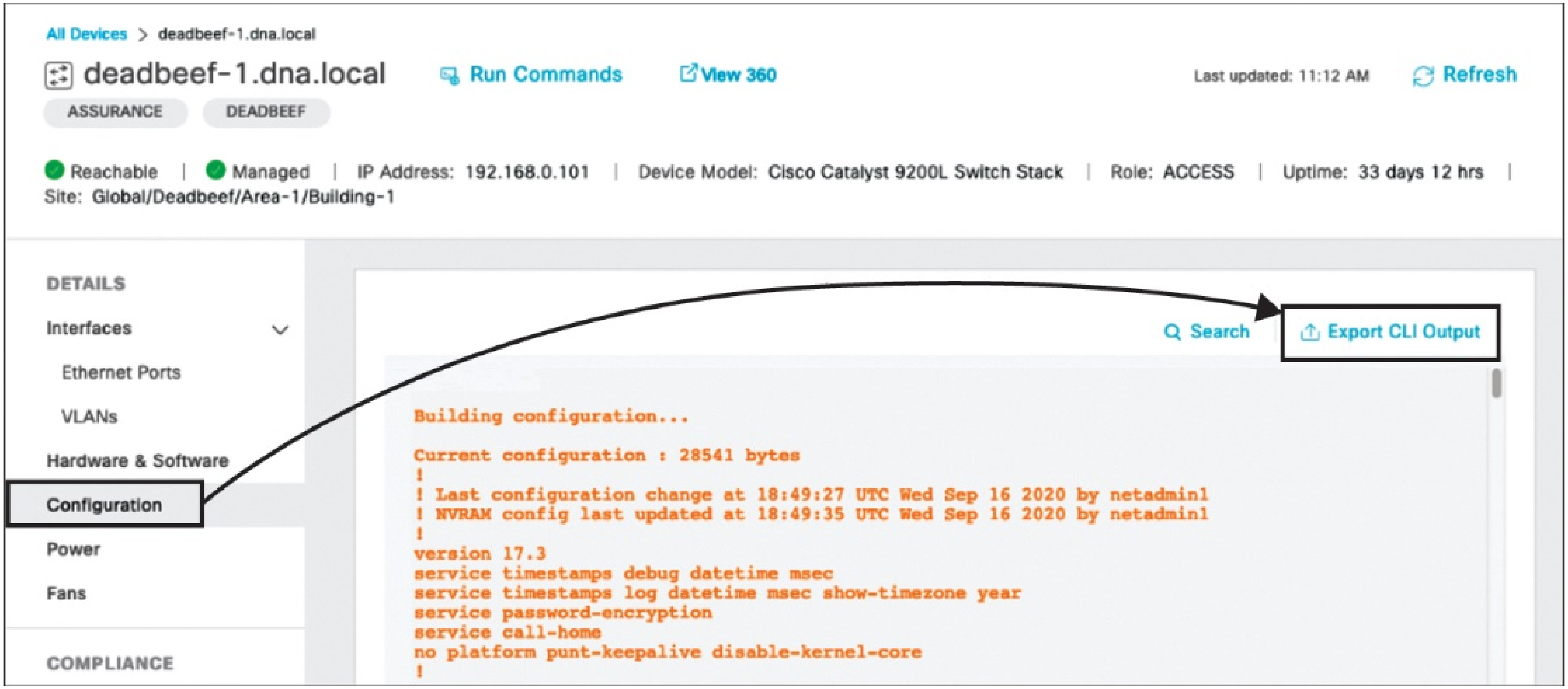


Figure 15-1 The Export CLI Output allows you to export the configuration of a device

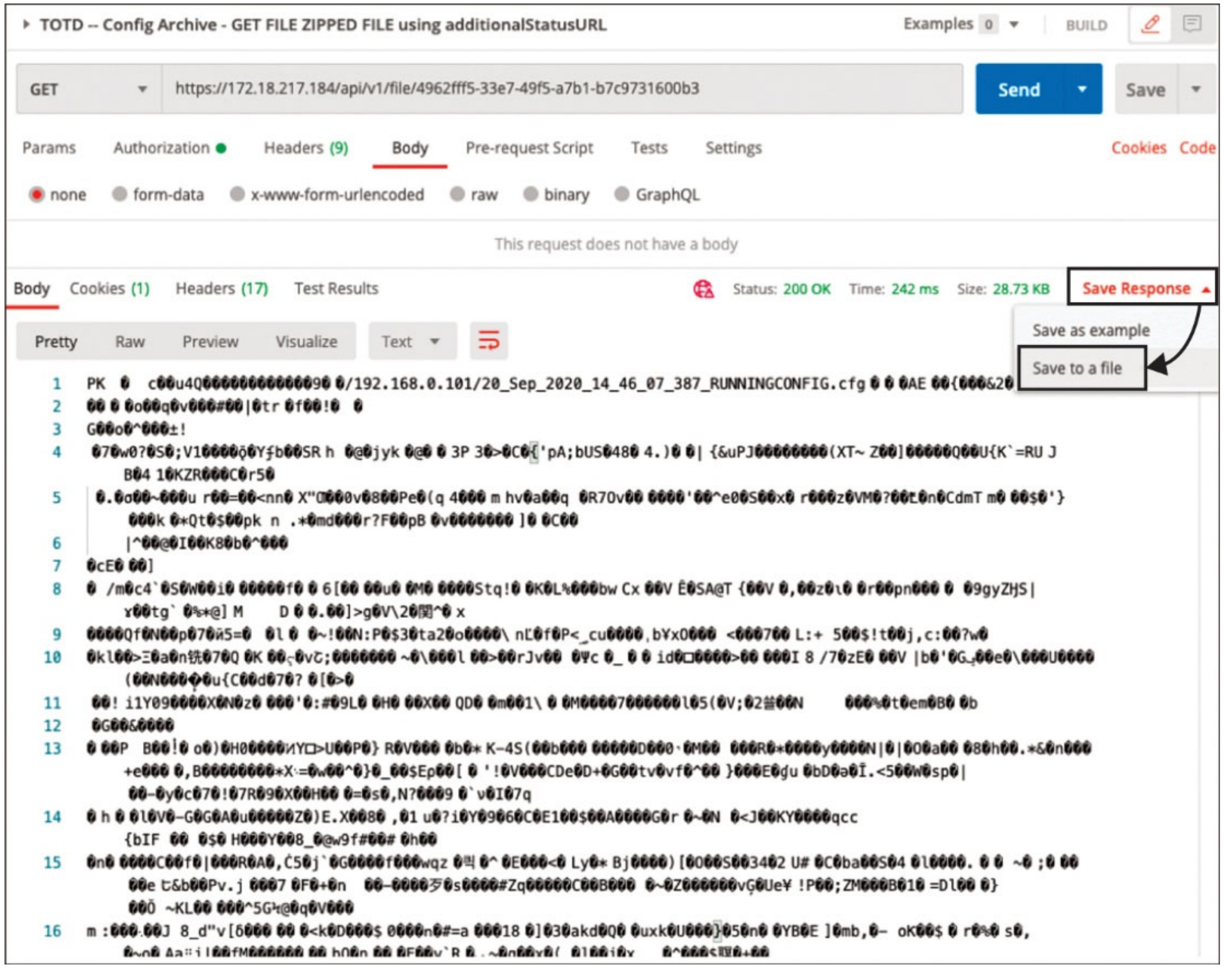


Figure 15-2 Saving the file containing the configurations through API calls

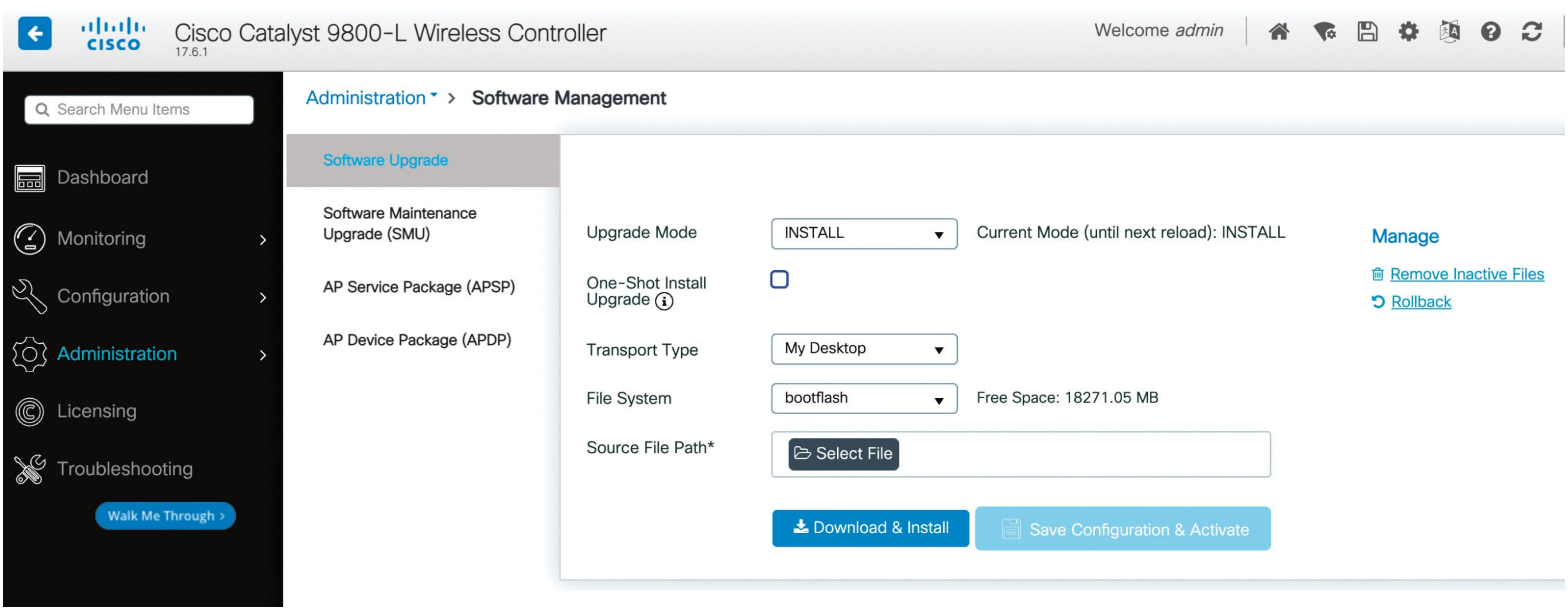


Figure 15-3 The Catalyst 9800 upgrade page

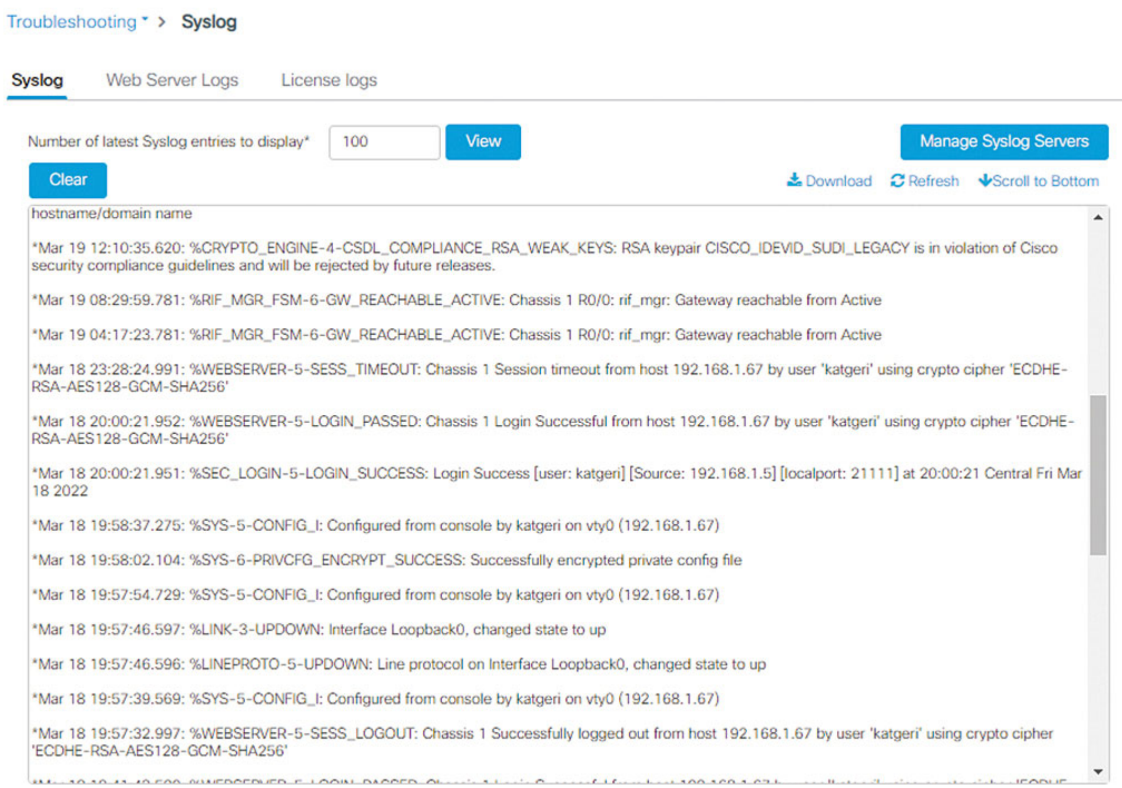


Figure 16-1 Troubleshooting Dashboard (Syslog)

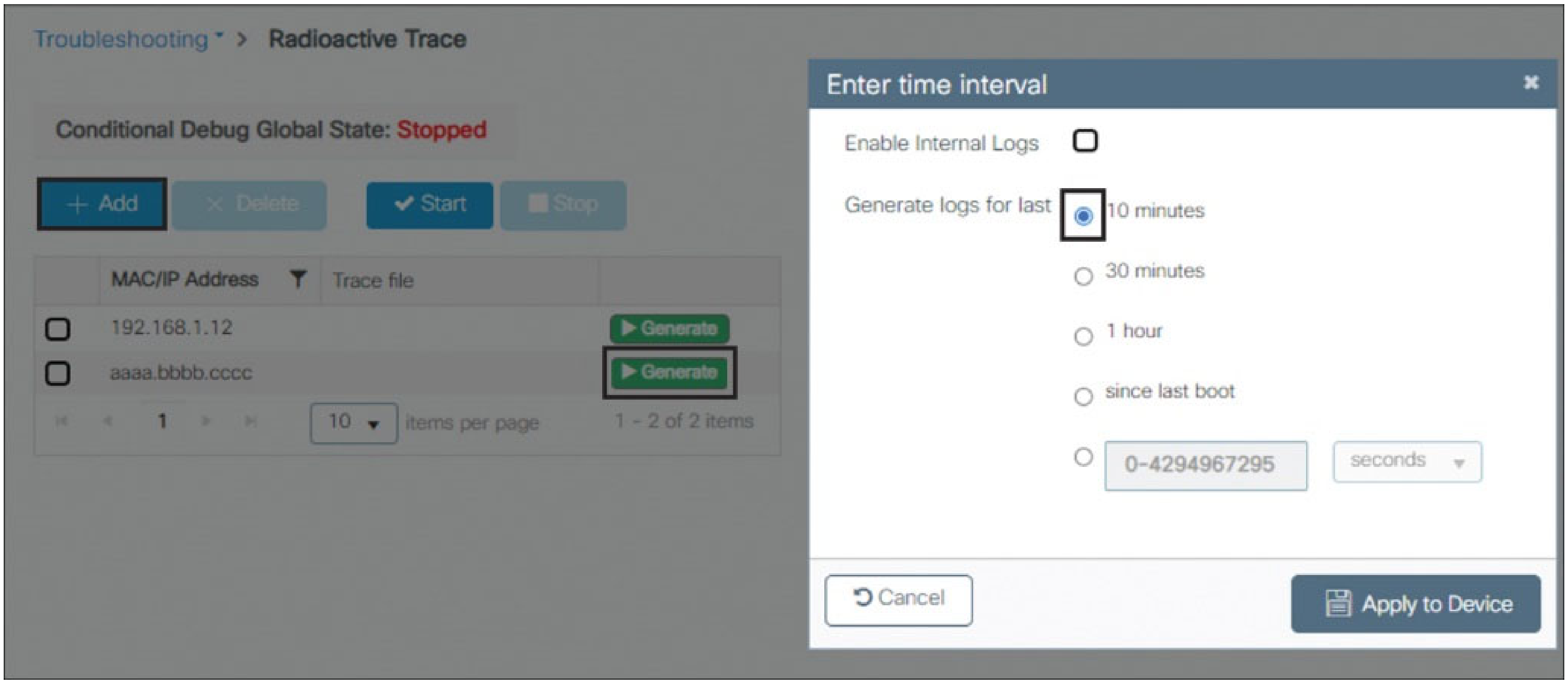


Figure 16-2 Collecting always-on traces for a client MAC from the C9800 GUI

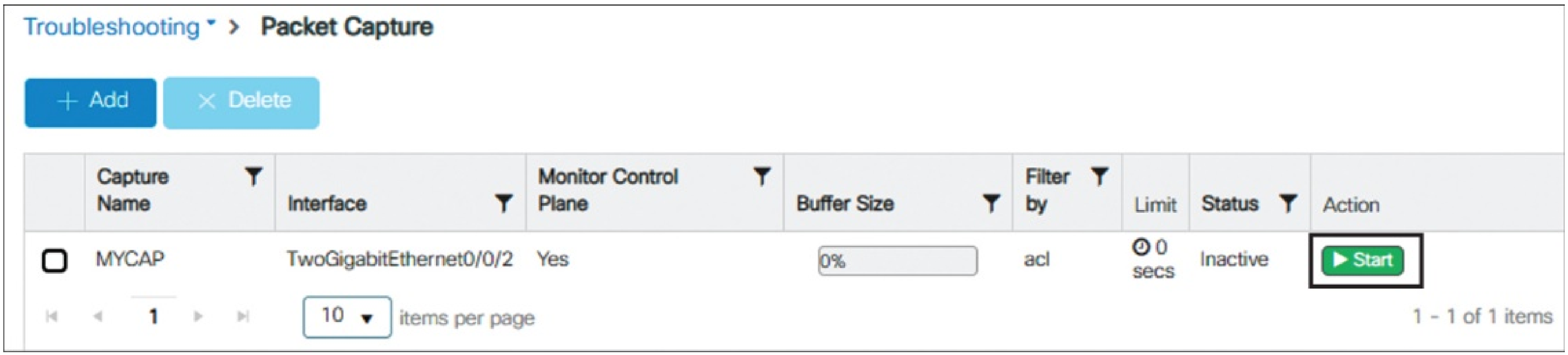


Figure 16-3 Selecting the AP MAC address for conditional tracing from APs listed under Monitoring

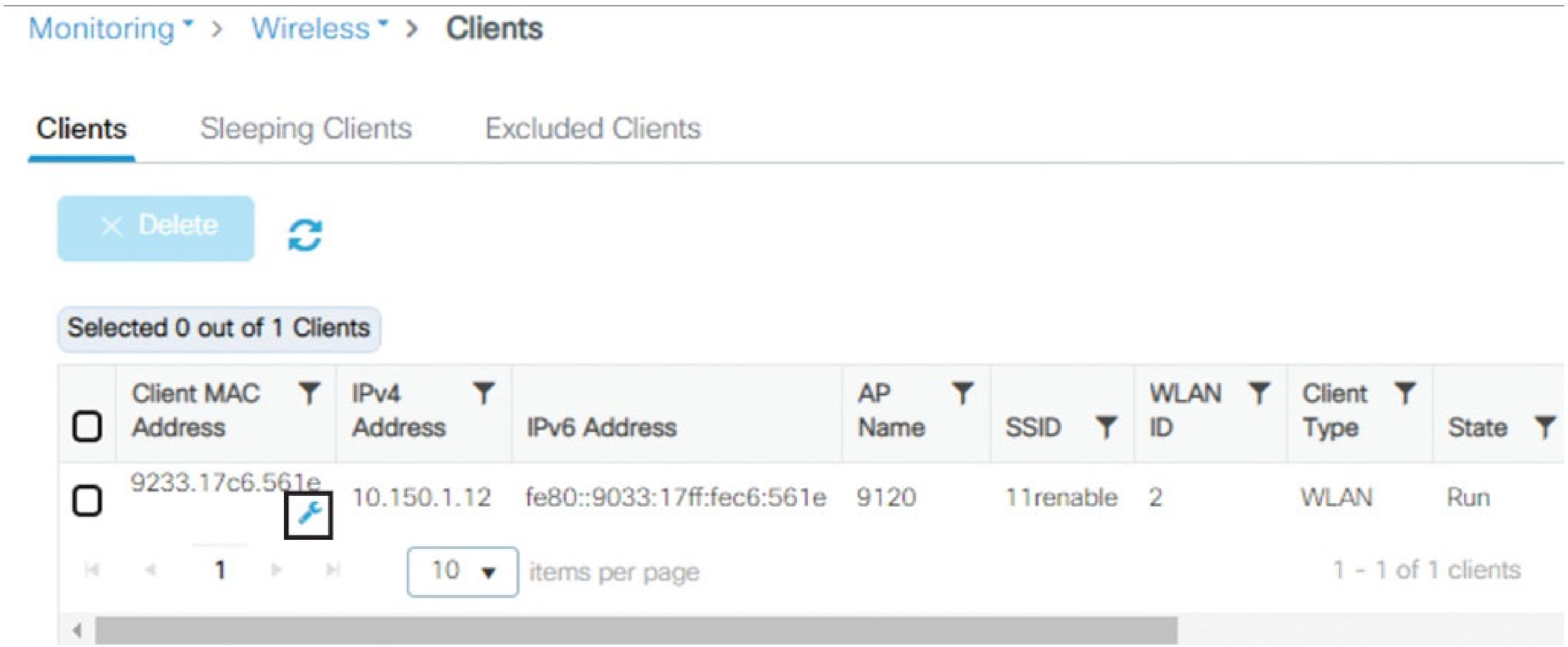


Figure 16-4 Selecting the client MAC address for conditional tracing from the client listed under Monitoring

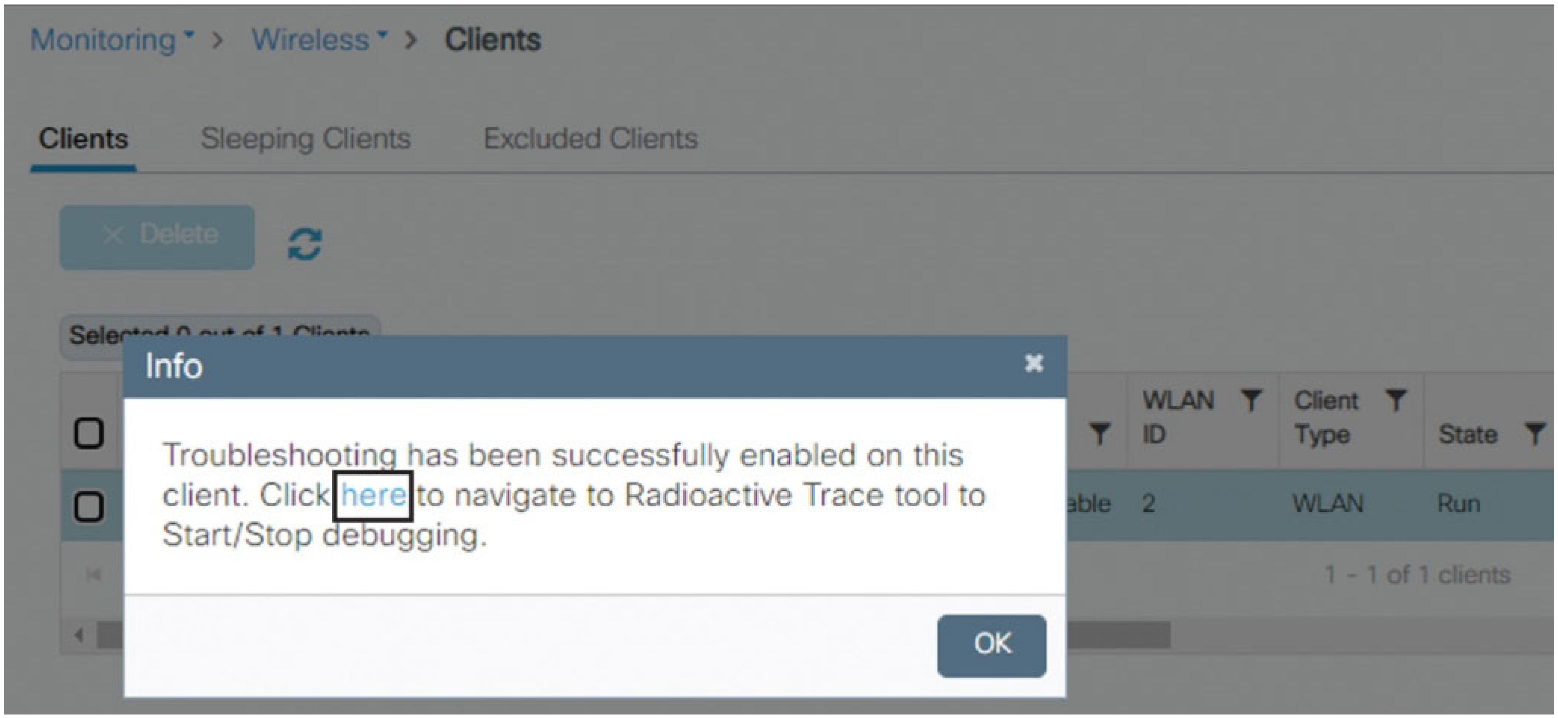


Figure 16-5 Direct navigation of the C9800 GUI from Monitoring to Troubleshooting

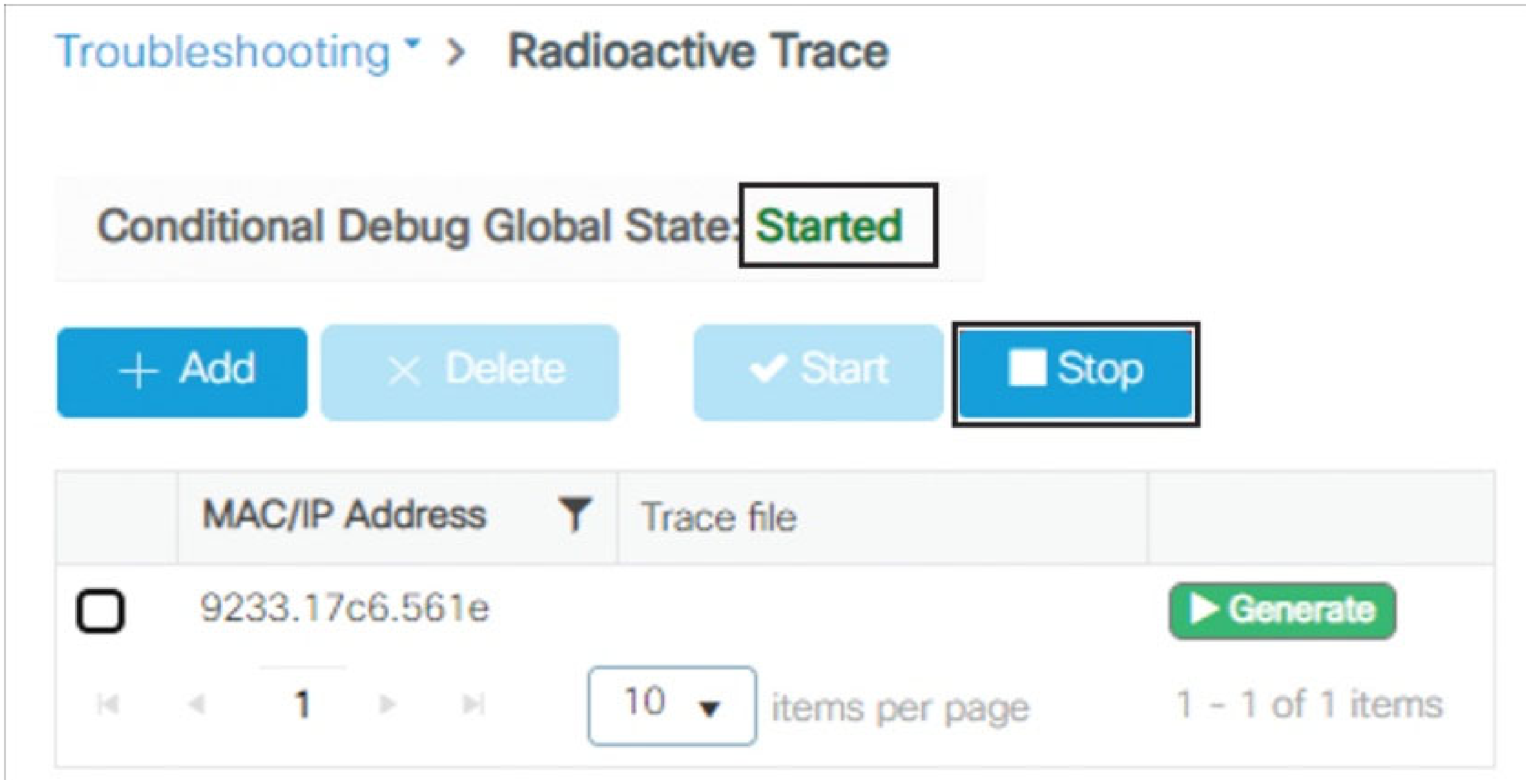


Figure 16-6 Starting a radioactive trace on the C9800 GUI

From *Understanding and Troubleshooting Cisco Catalyst 9800 Series Wireless Controller by* Simone Arena, Francisco Sedano Crippa, Nicolas Darchis and Sudha Katgeri(9780137492329) Cisco Press

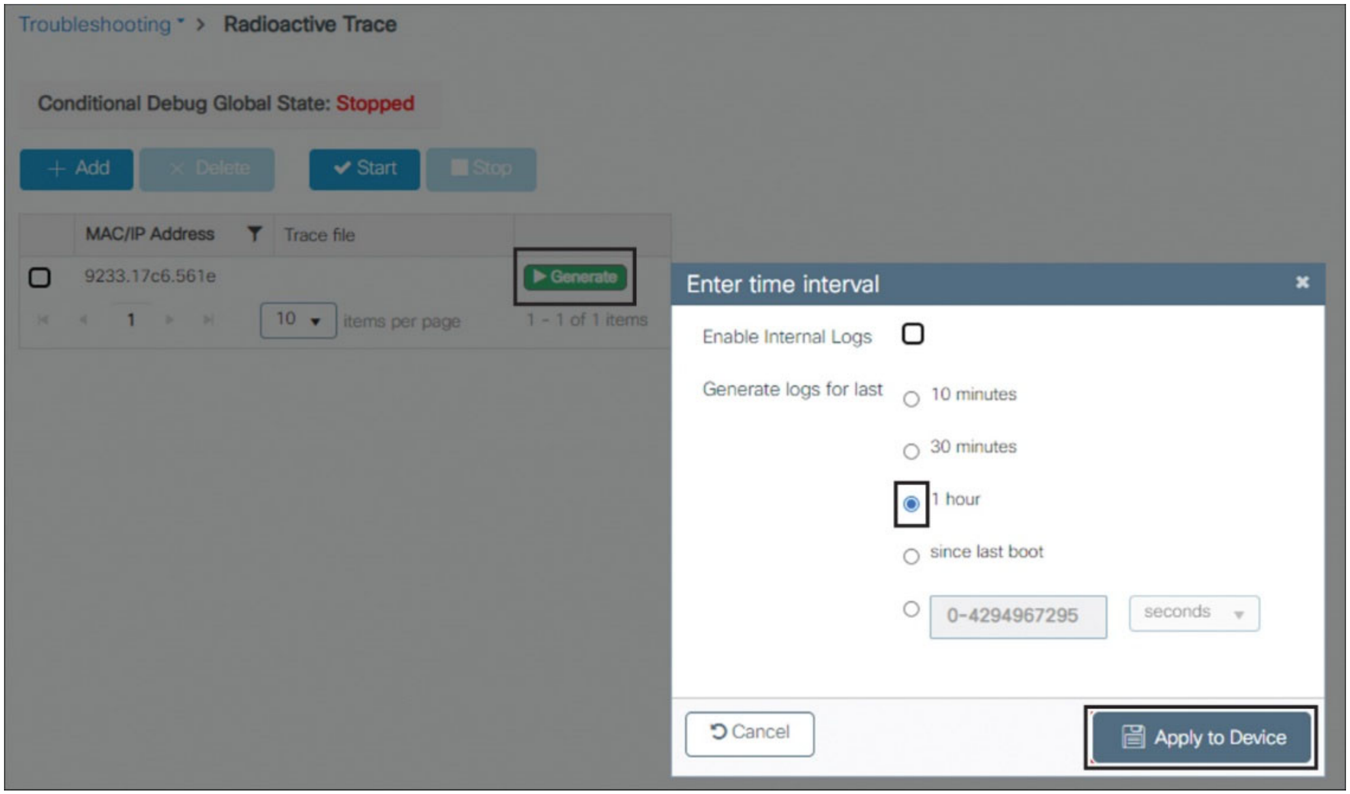


Figure 16-7 Collecting customer-use radioactive traces from the C9800 GUI

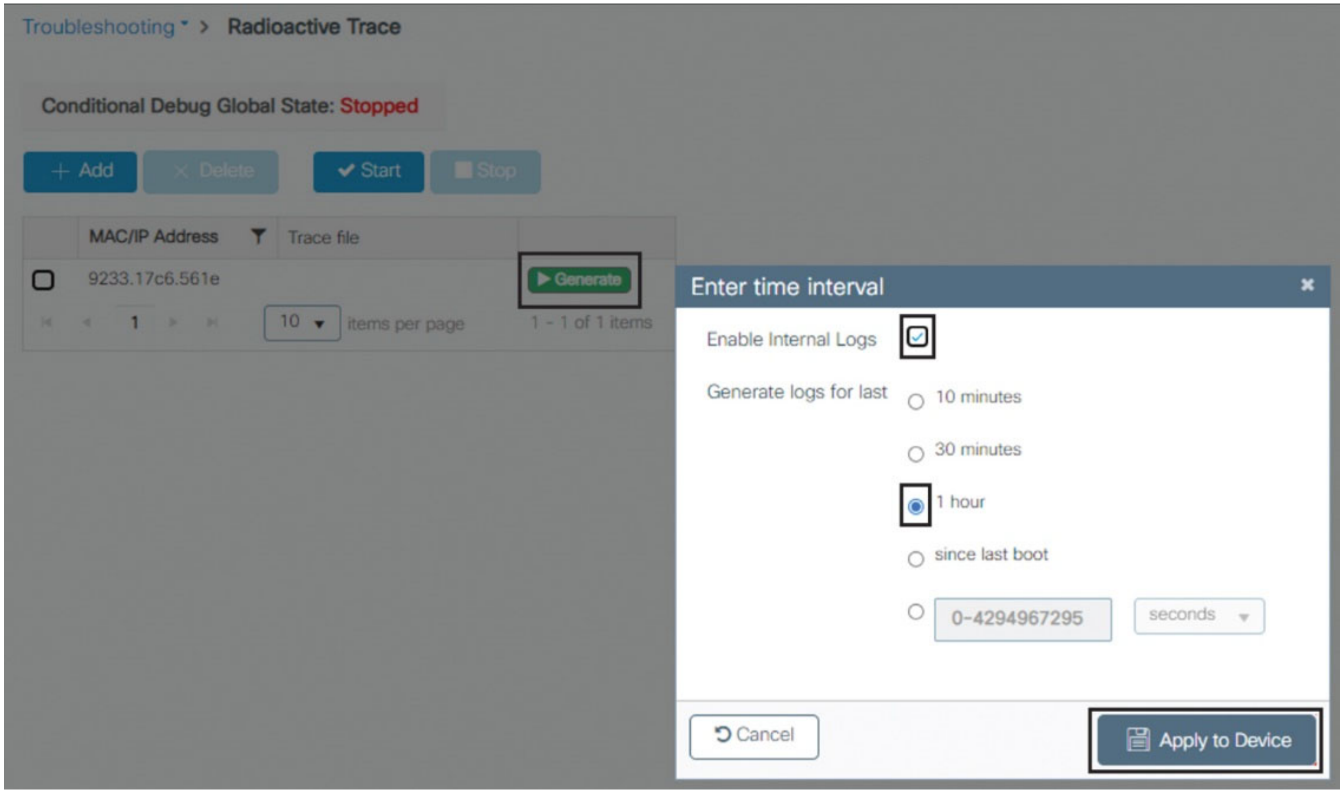


Figure 16-8 Collecting internal radioactive traces from the C9800 GUI

```
C9800#show platform software punt-policer | include Punt|Cause|--|EPC
Per Punt-Cause Policer Configuration and Packet Counters
Punt  Description  Config Rate(pps)   Conform Packets    Dropped Packets     Config Burst(pkts)  Config Alert
Cause               Normal   High     Normal     High     Normal       High    Normal   High        Normal   High
--------------------------------------------------------------------------------------------------------------
 75     EPC         8738     1000       0          0        0            0      8738     1000       Off      Off
```

Figure 16-9 Viewing the default EPC punt policer

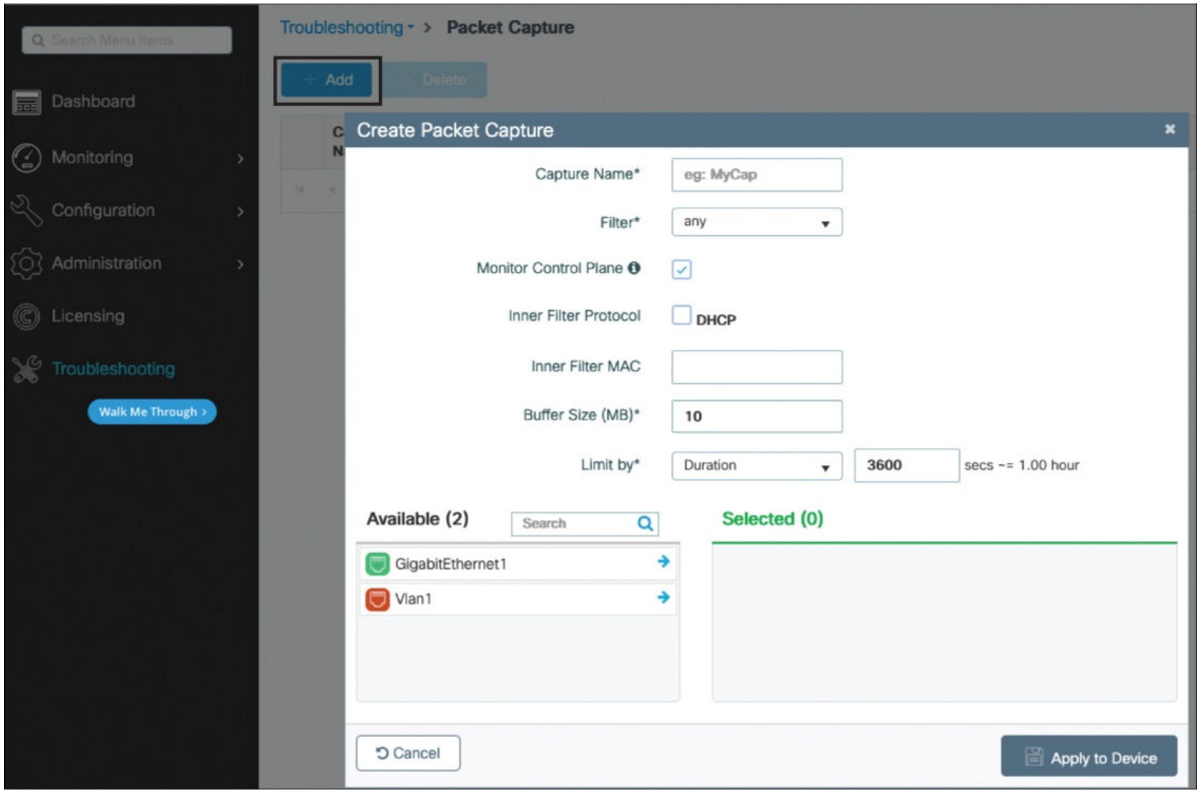
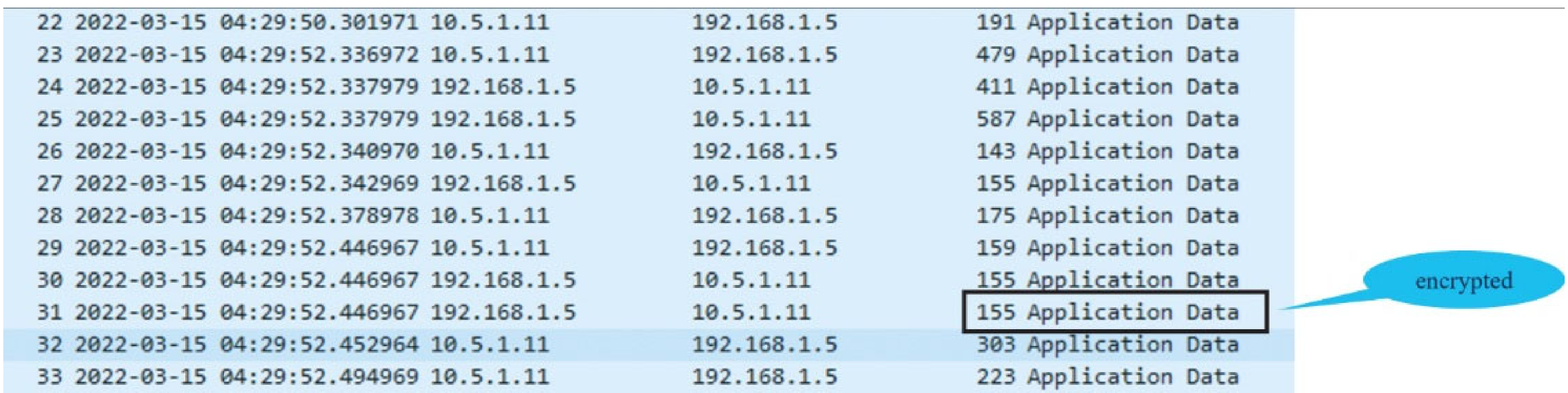


Figure 16-10 Enabling EPC on the C9800 Troubleshooting dashboard



22	2022-03-15 04:29:50.301971	10.5.1.11	192.168.1.5	191	Application Data
23	2022-03-15 04:29:52.336972	10.5.1.11	192.168.1.5	479	Application Data
24	2022-03-15 04:29:52.337979	192.168.1.5	10.5.1.11	411	Application Data
25	2022-03-15 04:29:52.337979	192.168.1.5	10.5.1.11	587	Application Data
26	2022-03-15 04:29:52.340970	10.5.1.11	192.168.1.5	143	Application Data
27	2022-03-15 04:29:52.342969	192.168.1.5	10.5.1.11	155	Application Data
28	2022-03-15 04:29:52.378978	10.5.1.11	192.168.1.5	175	Application Data
29	2022-03-15 04:29:52.446967	10.5.1.11	192.168.1.5	159	Application Data
30	2022-03-15 04:29:52.446967	192.168.1.5	10.5.1.11	155	Application Data
31	2022-03-15 04:29:52.446967	192.168.1.5	10.5.1.11	155	Application Data
32	2022-03-15 04:29:52.452964	10.5.1.11	192.168.1.5	303	Application Data
33	2022-03-15 04:29:52.494969	10.5.1.11	192.168.1.5	223	Application Data

Figure 16-11 EPC showing only encrypted packets with data DTLS without control plane monitoring

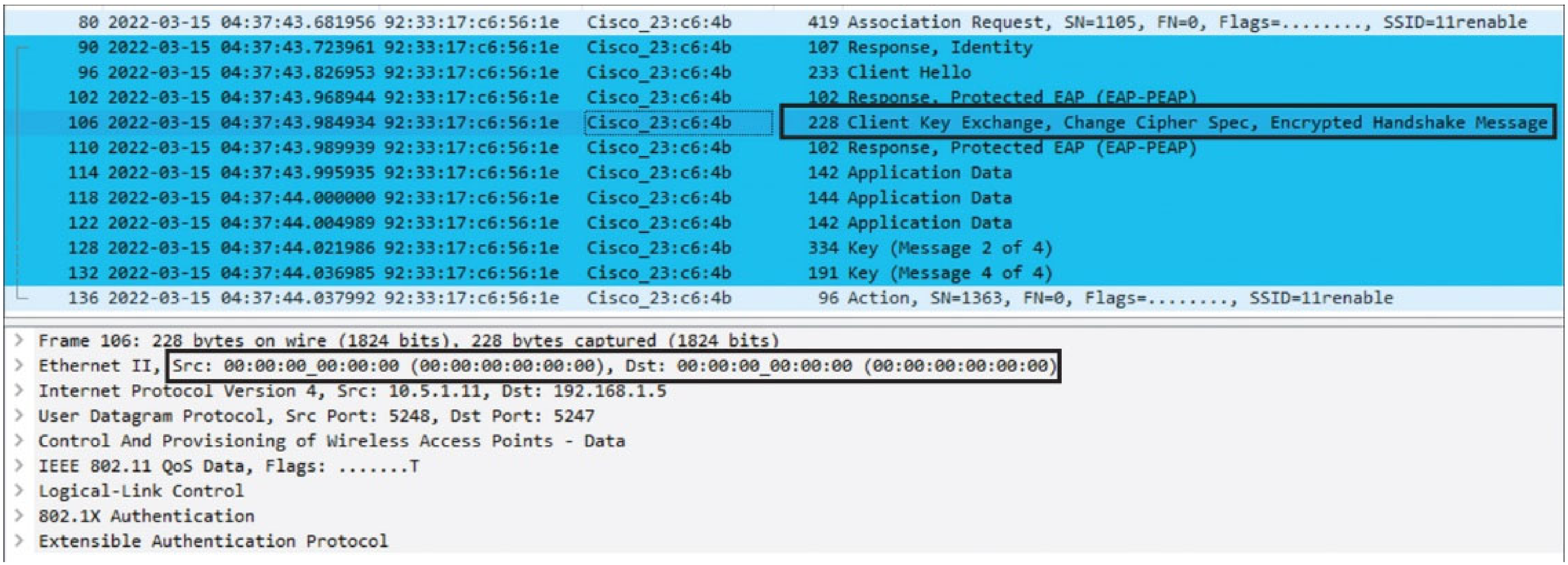


Figure 16-12 EPC showing decrypted packets even with data DTLS enabled with control plane monitoring

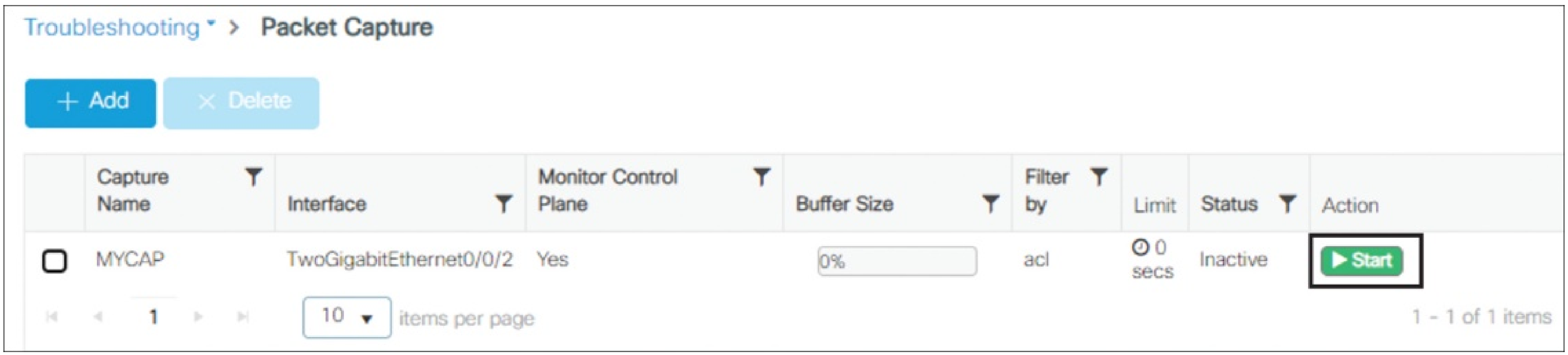


Figure 16-13 Starting EPC on the C9800 Troubleshooting dashboard

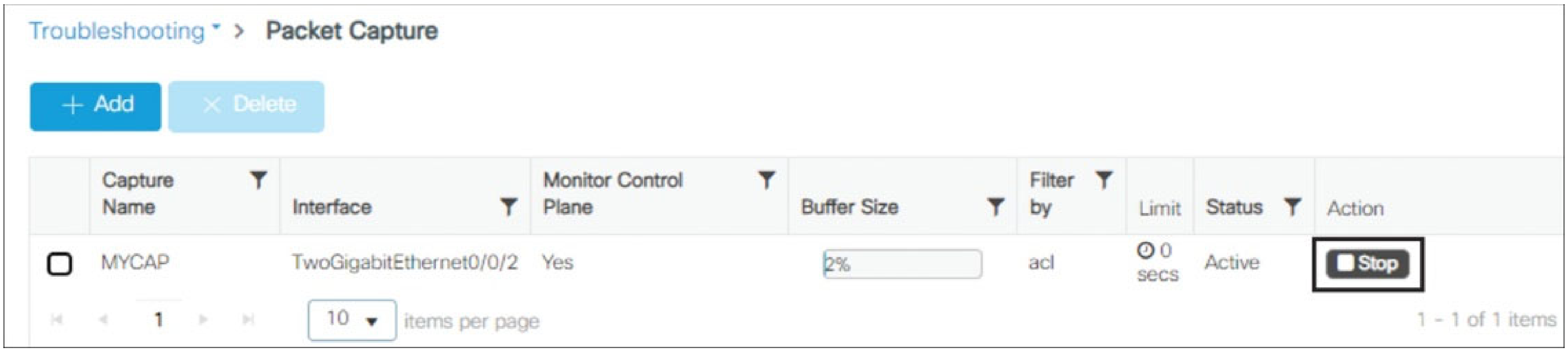


Figure 16-14 Monitoring progress and stopping EPC capture on the C9800 Troubleshooting dashboard

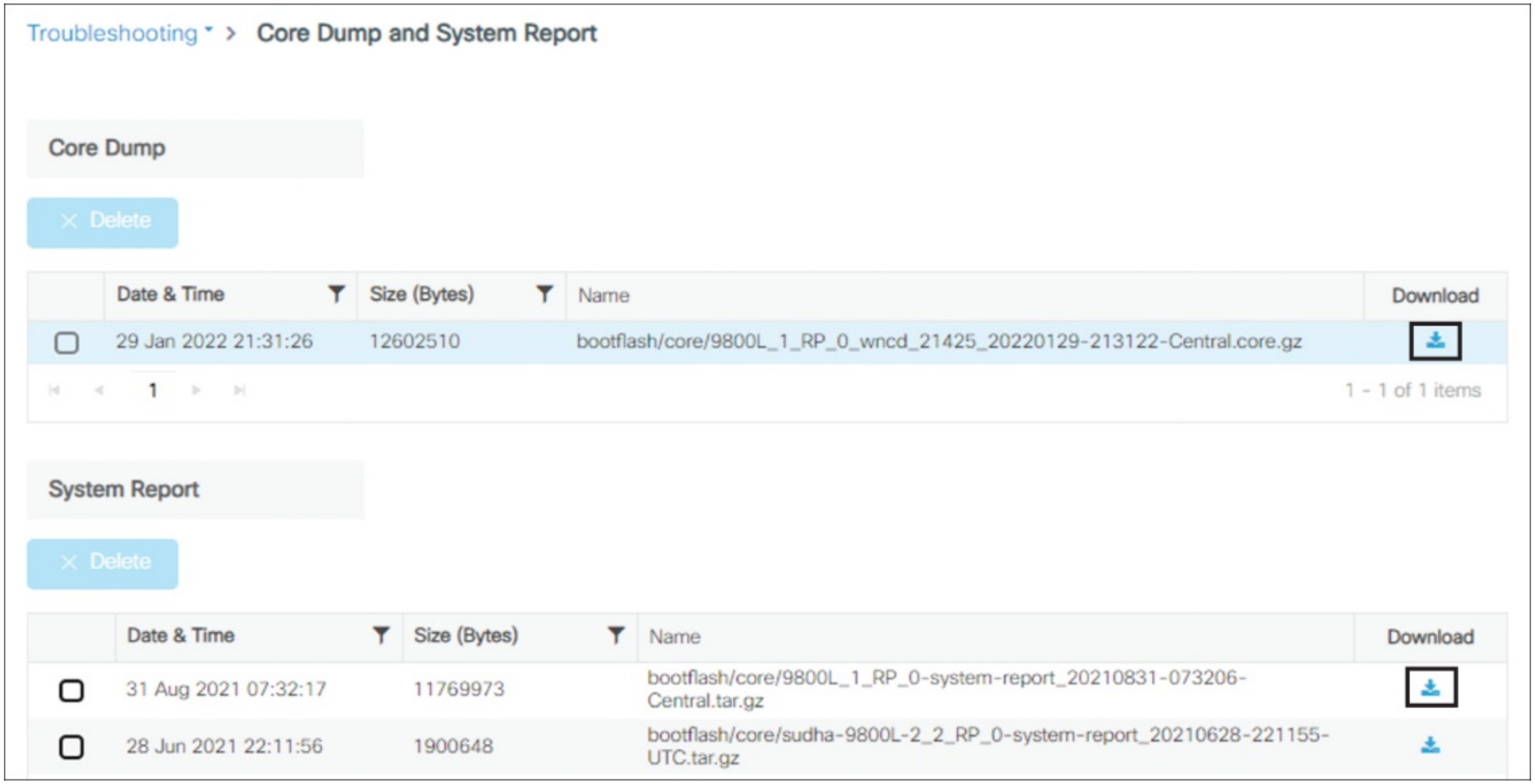


Figure 16-15 Core Dump and System Report web page that shows the WNCd core and two older system reports

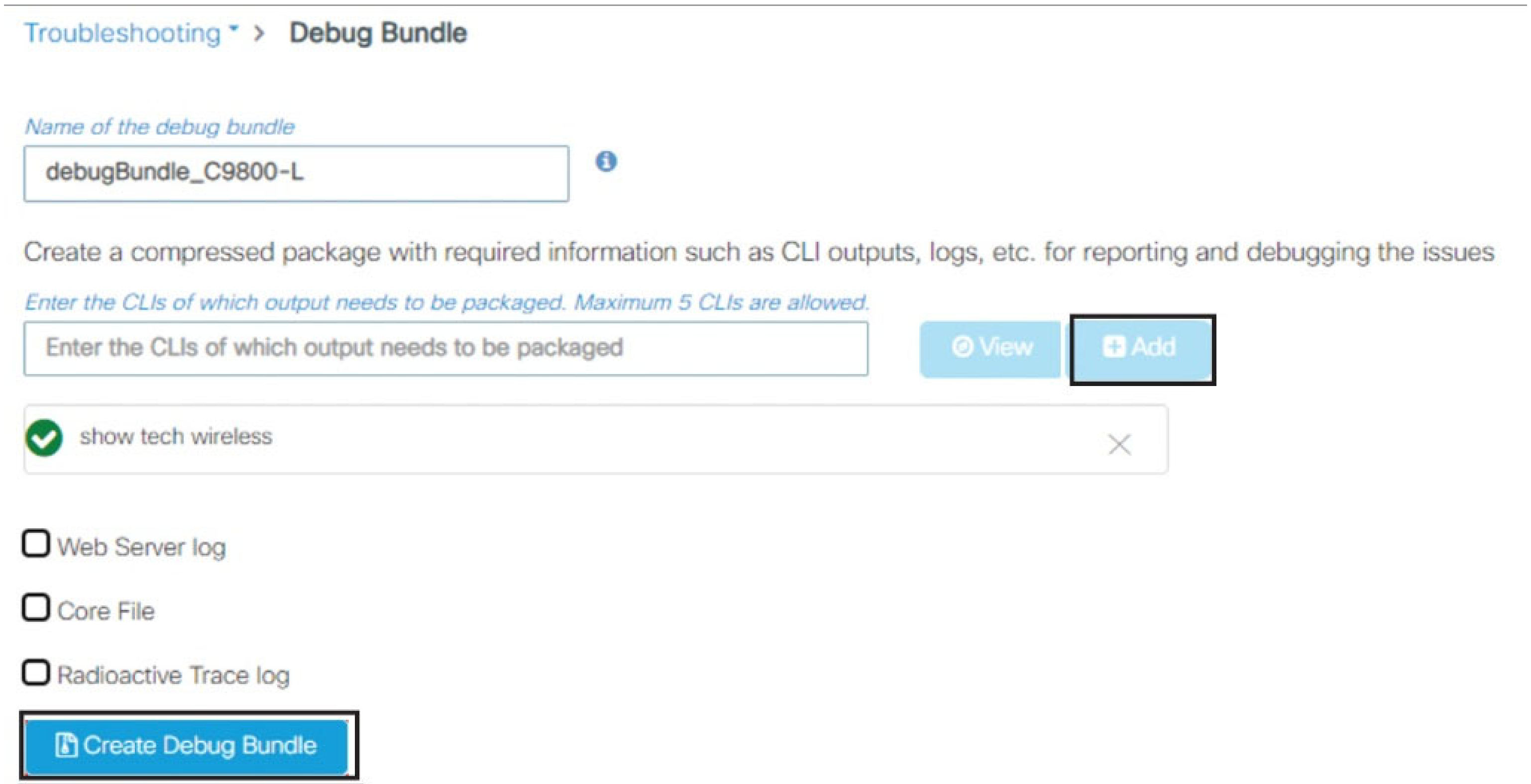


Figure 16-16 Troubleshooting the dashboard—Debug Bundle

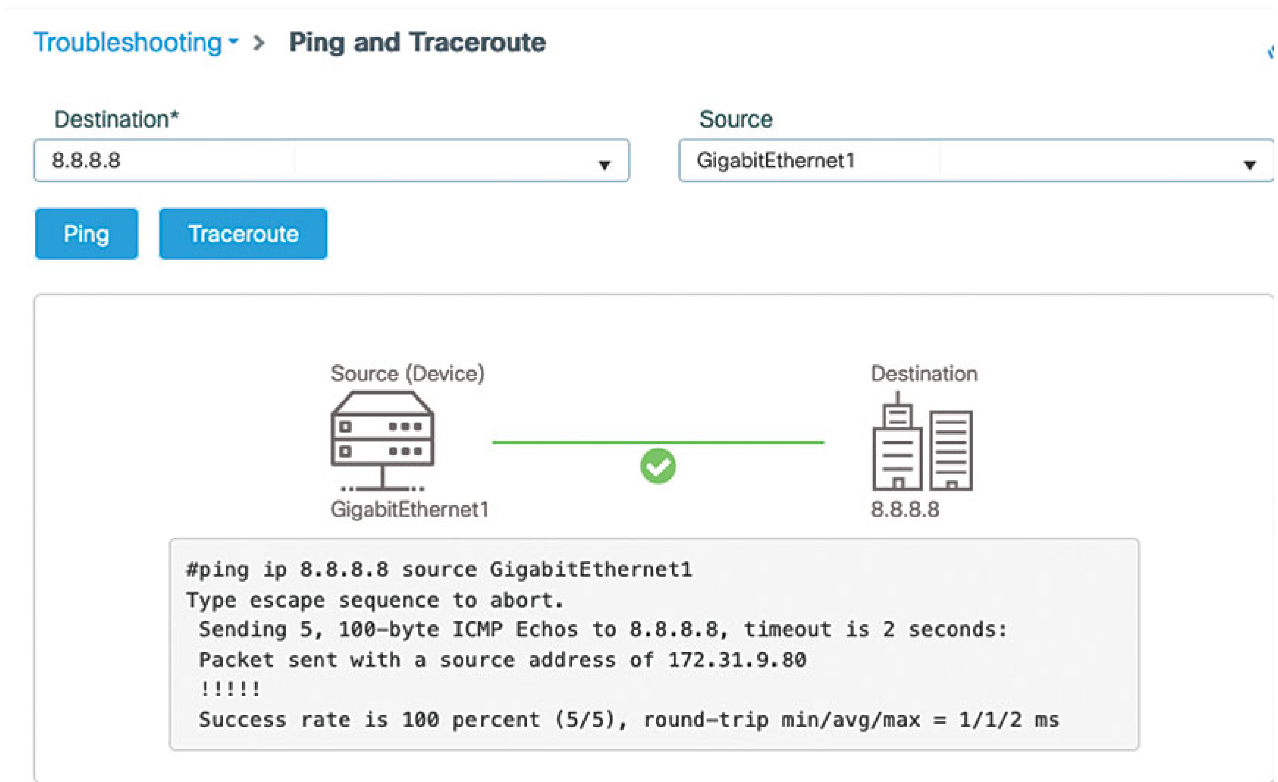


Figure 16-17 Troubleshooting the dashboard—Ping and Trace Route

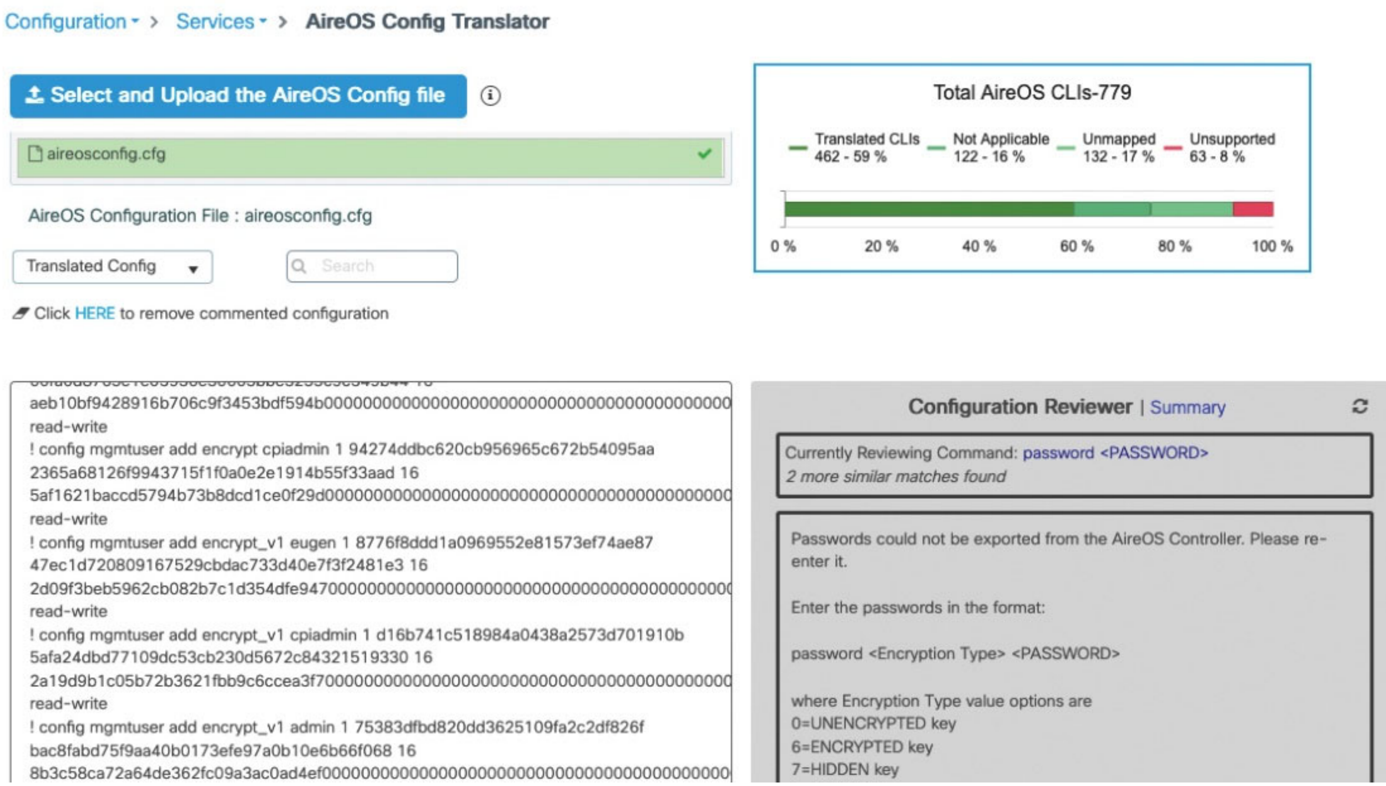


Figure 16-18 AireOS Config Translator

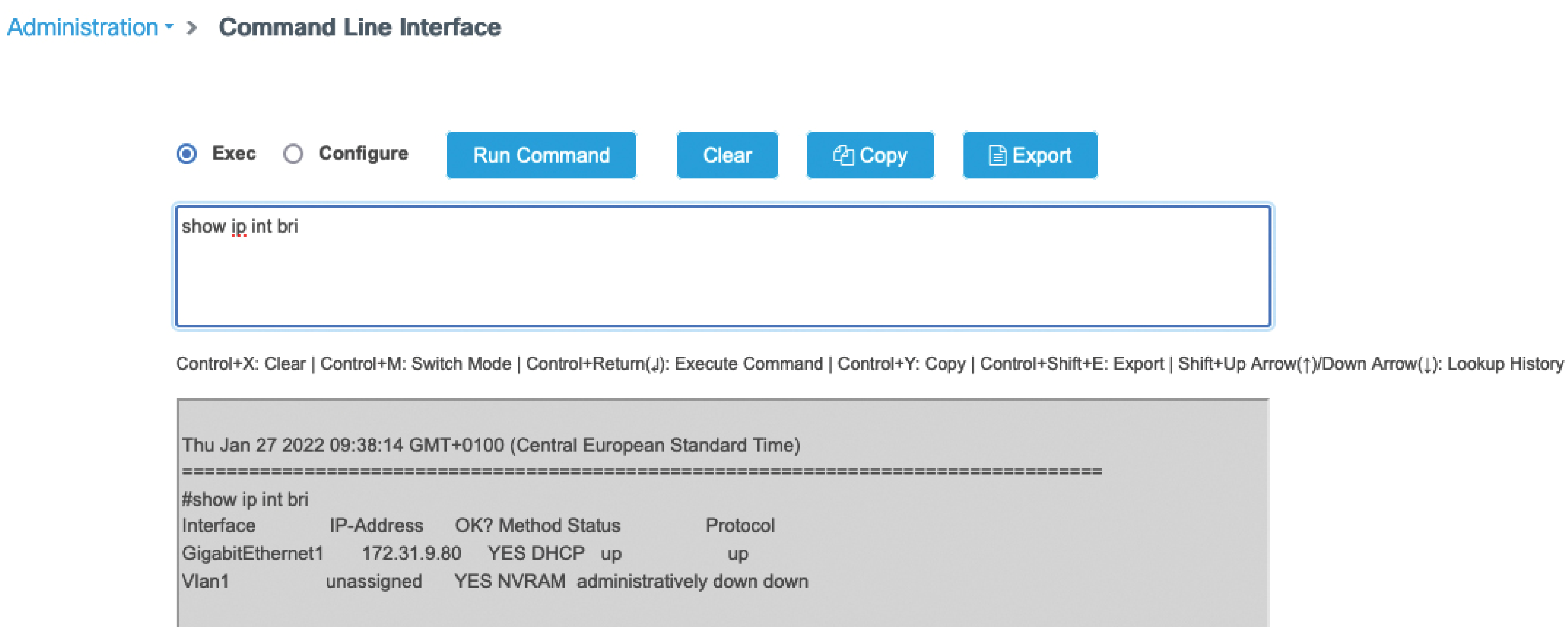


Figure 16-19 Command-line interface on the C9800 GUI

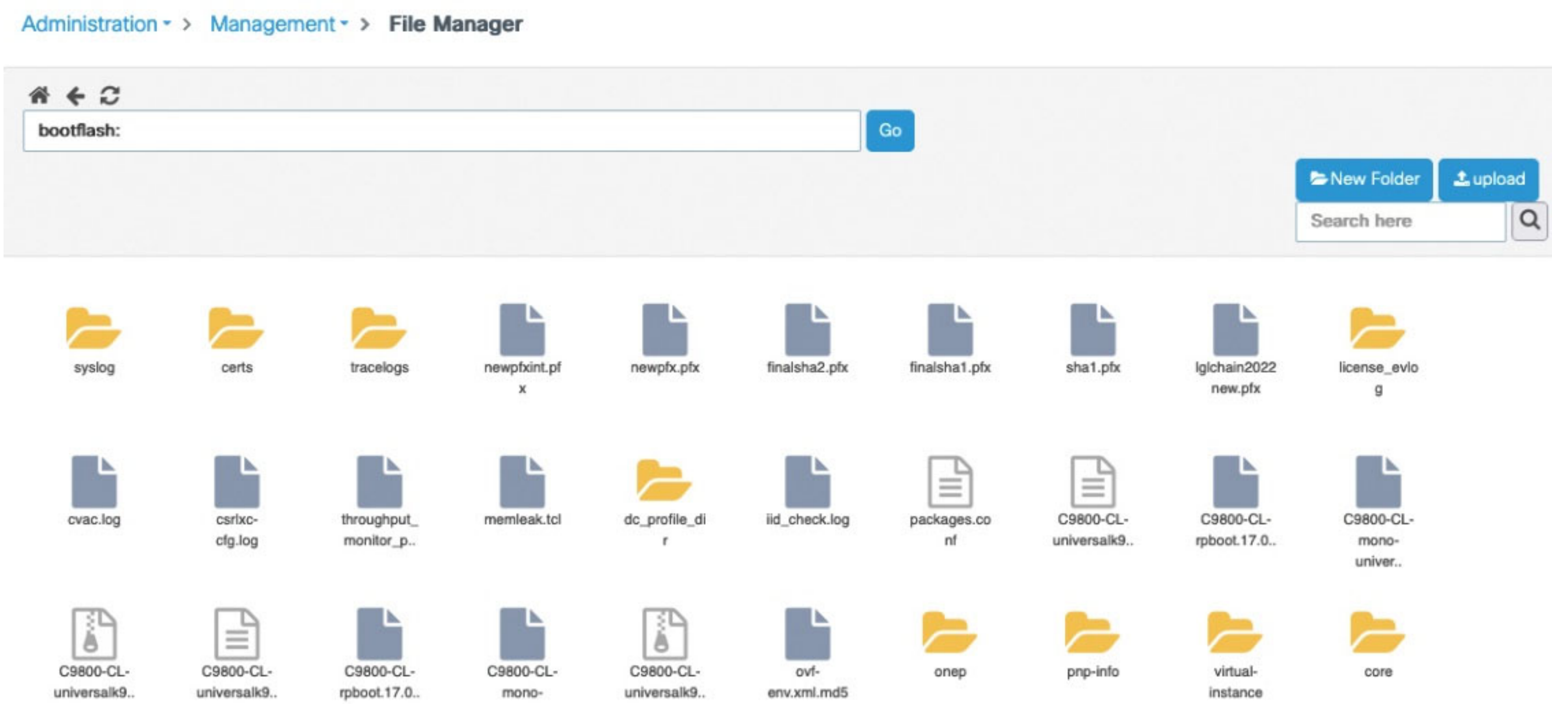


Figure 16-20 File Manager on the C9800 GUI

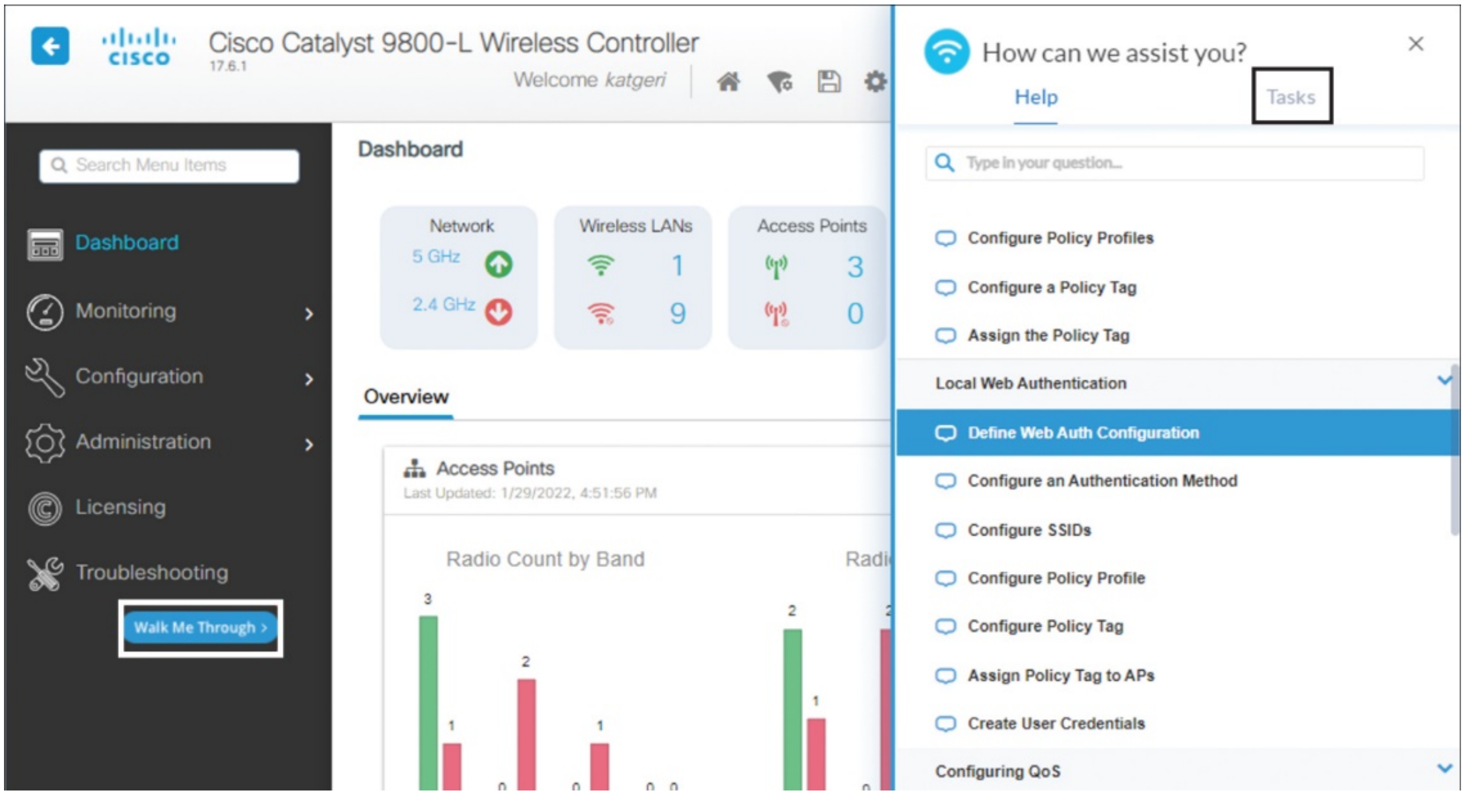


Figure 16-21 Configuration-guided workflow via Walk-me integration on the C9800 GUI

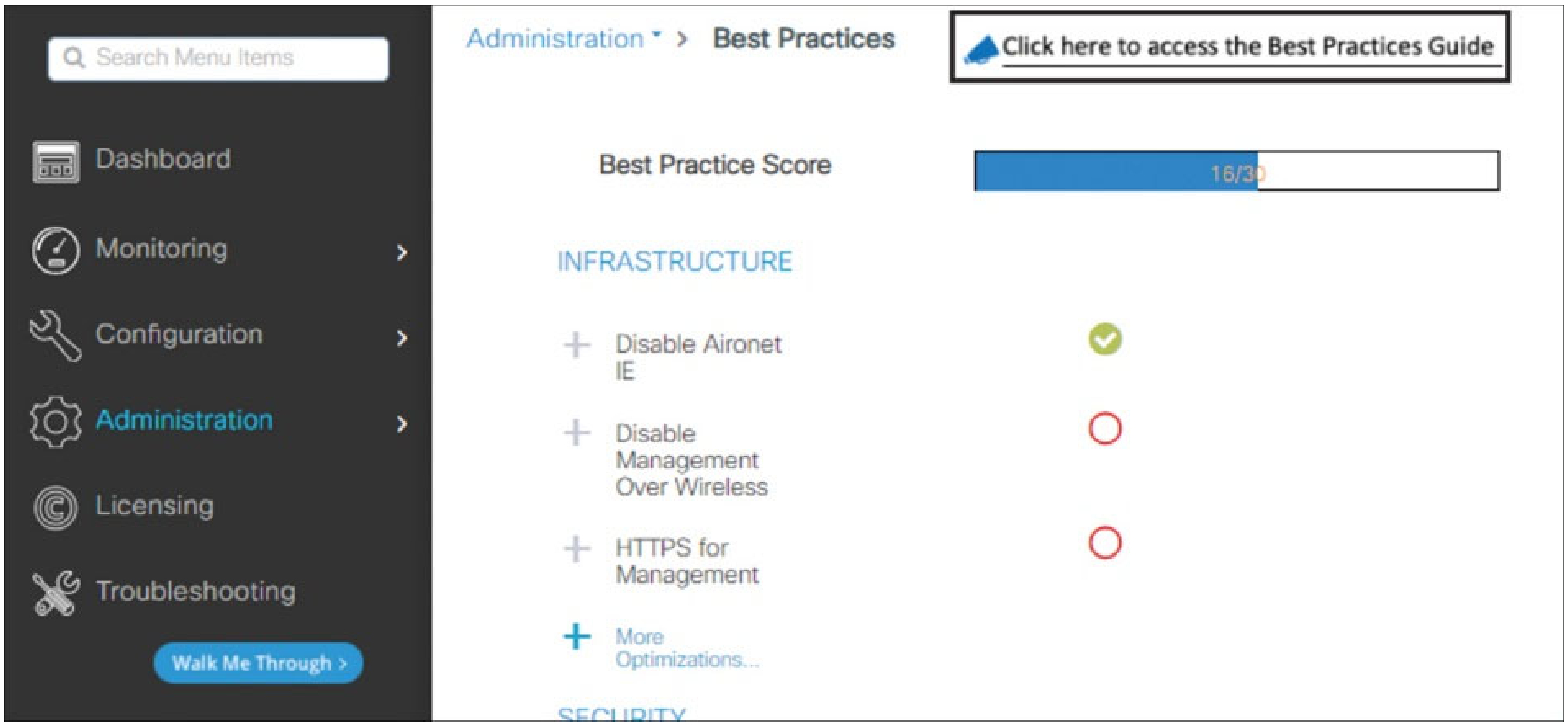


Figure 16-22 Best practices linked to the C9800 GUI via Walk-me integration

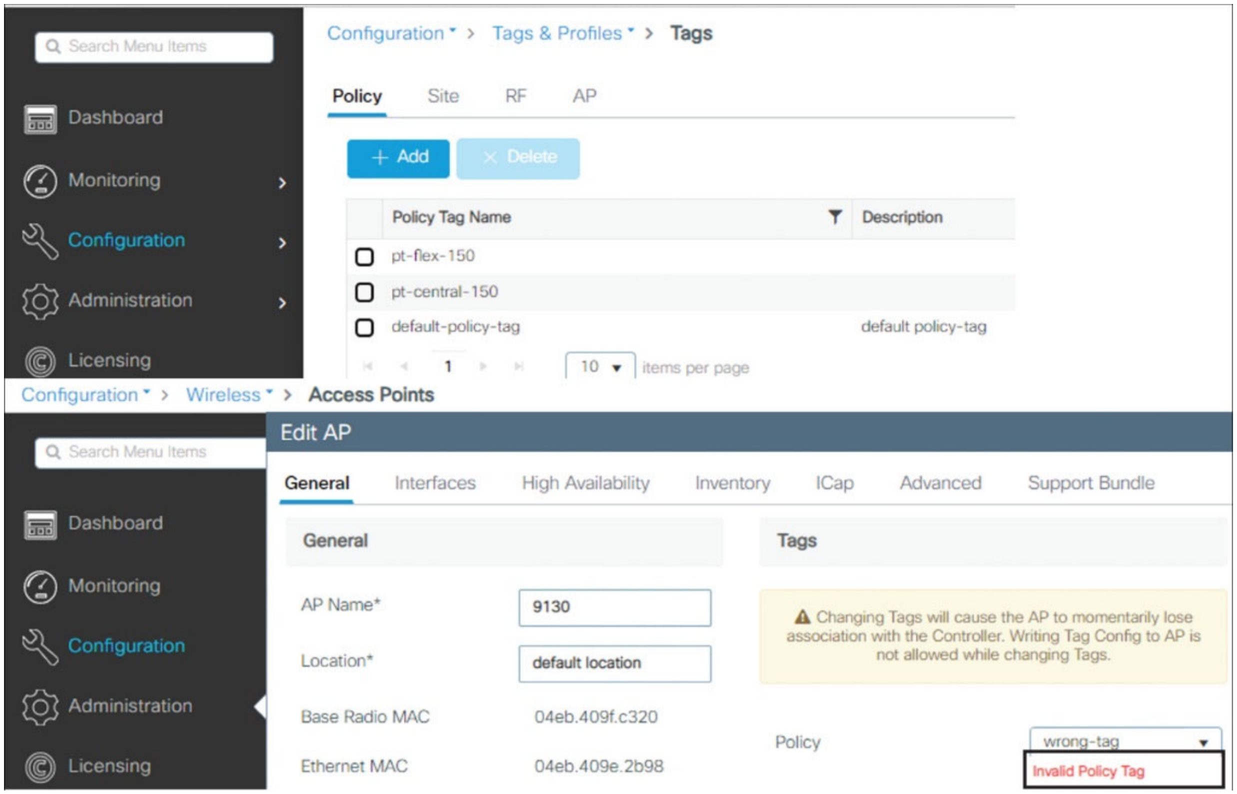


Figure 16-23 C9800 GUI preventing misconfiguration via configuration validation

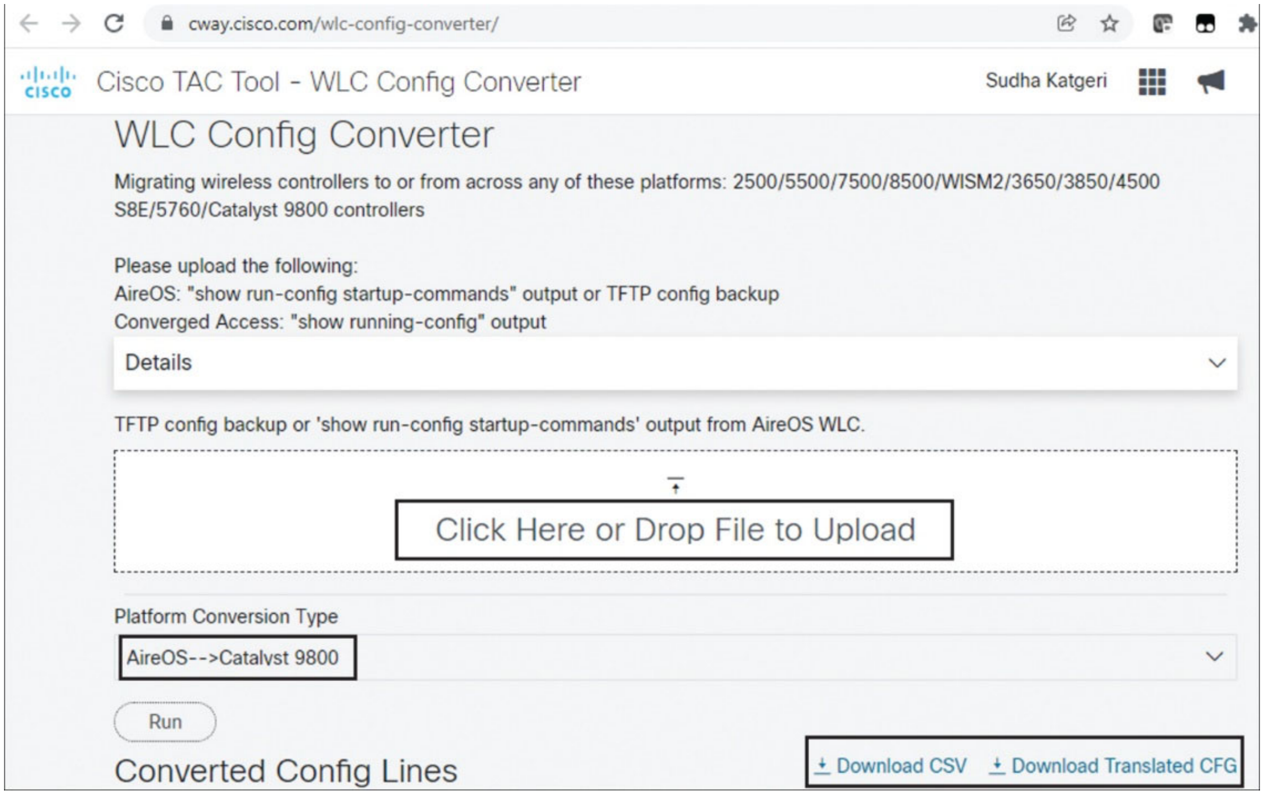


Figure 16-24 Wireless Config Convertor on cisco.com

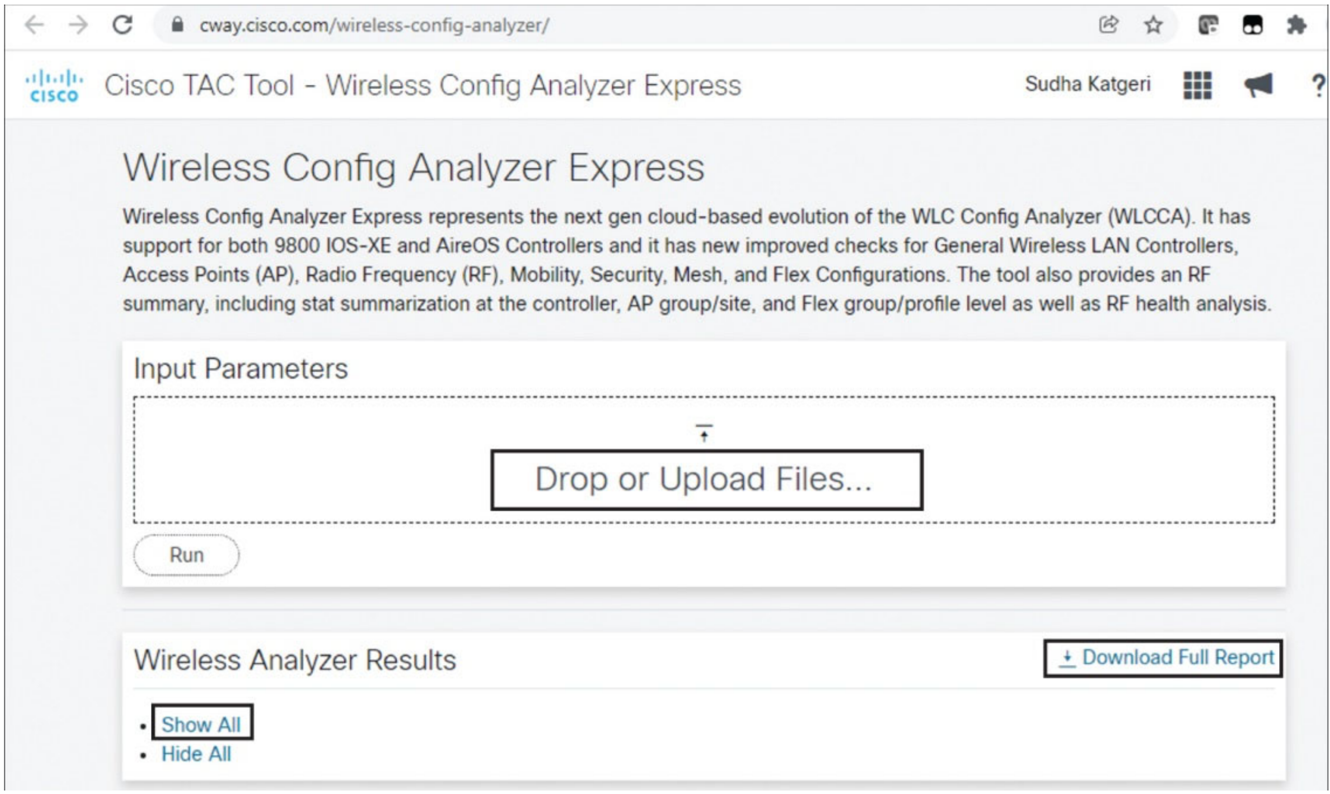


Figure 16-25 Wireless Config Analyzer Express on cisco.com

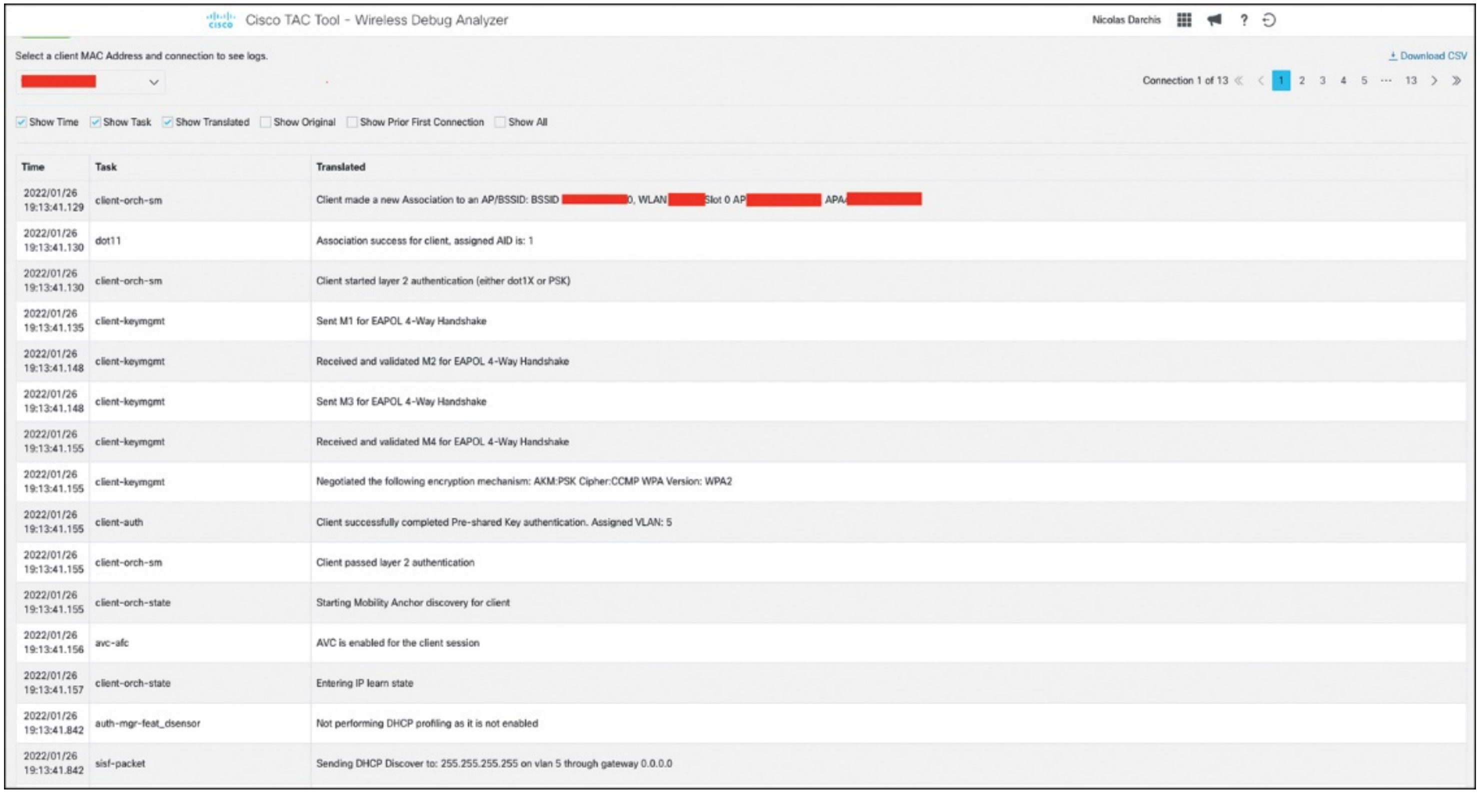


Figure 16-26 Wireless Debug Analyzer on cisco.com

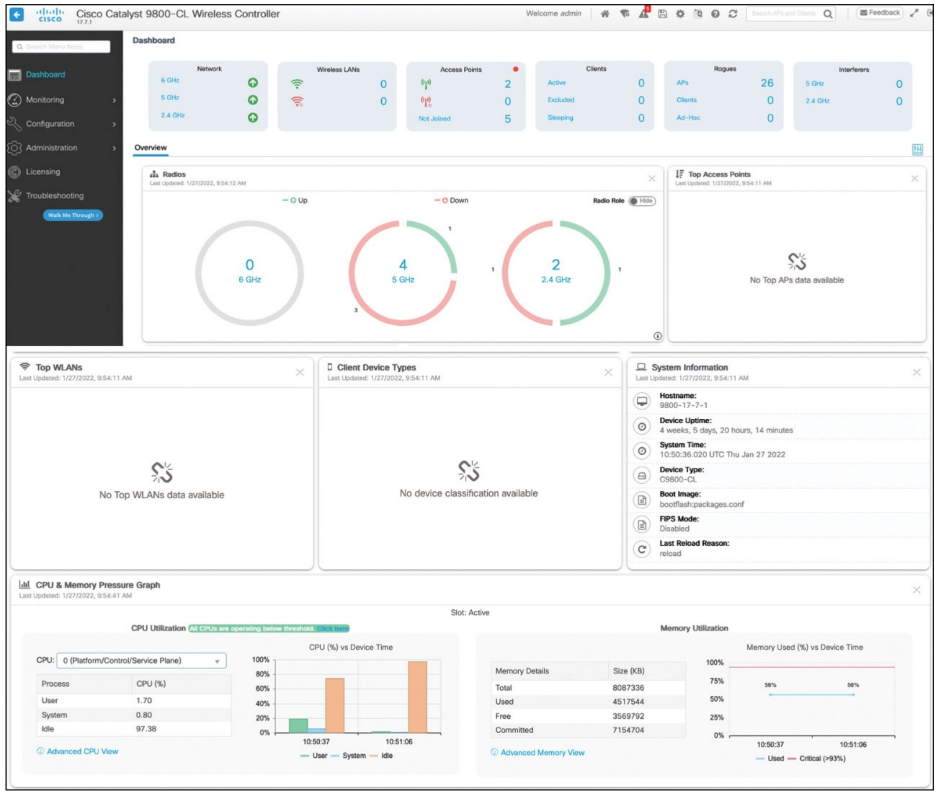


Figure 16-27 Dashboard on the C9800 GUI in IOS-XE 17.7.1

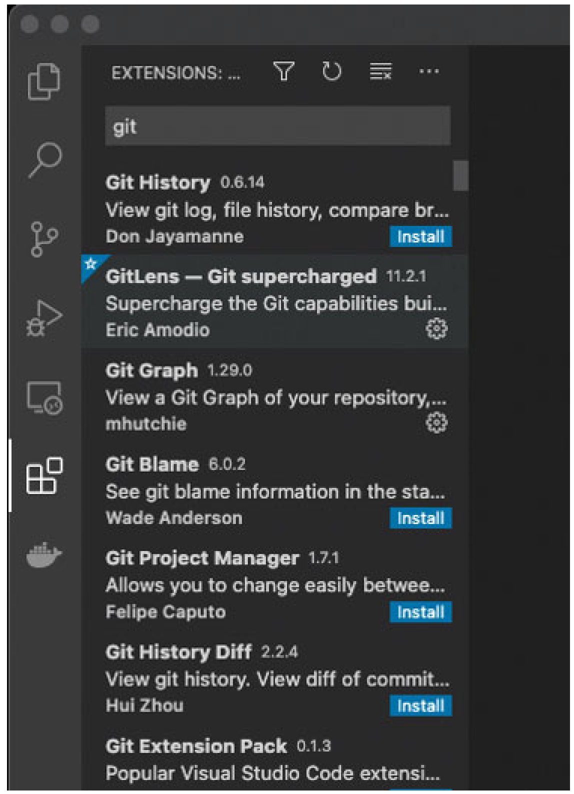


Figure A-1 Git plug-ins for Visual Studio Code

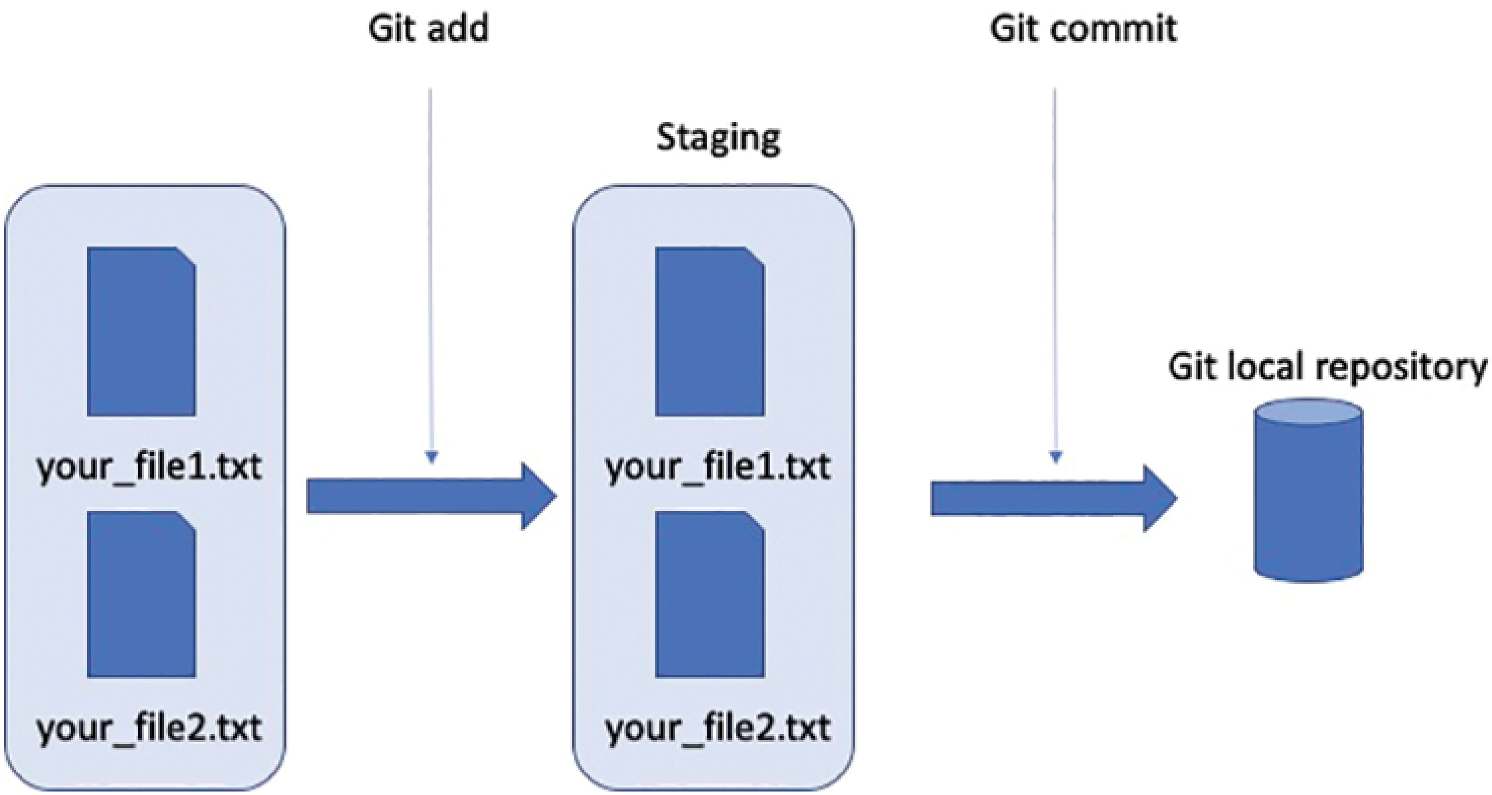


Figure A-2 Git workflow

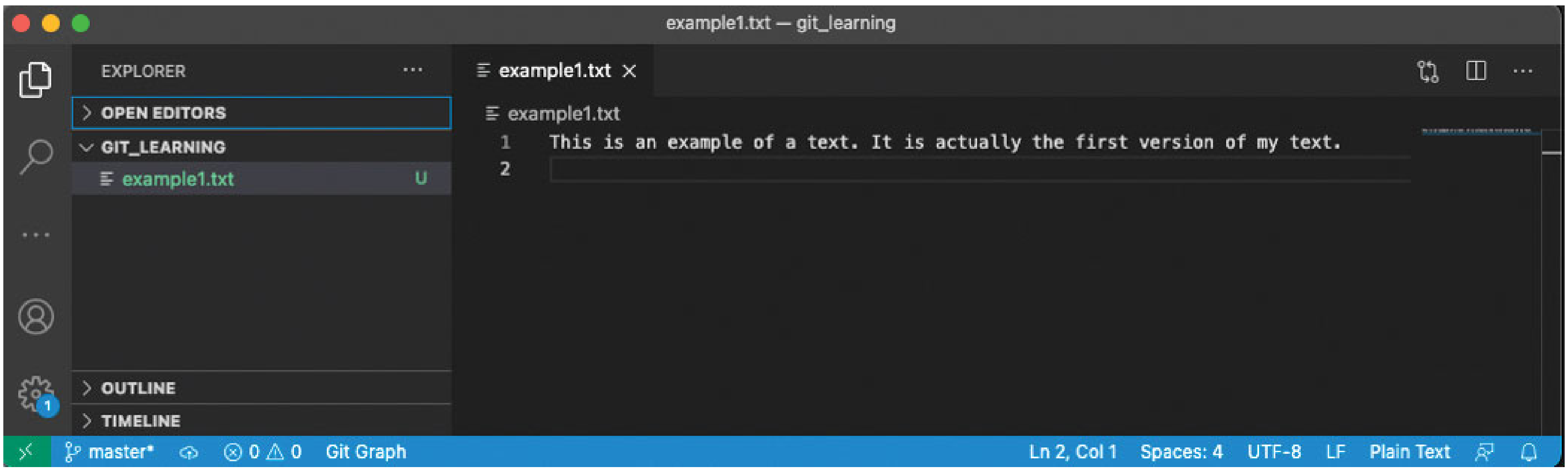


Figure A-3 Creating first file

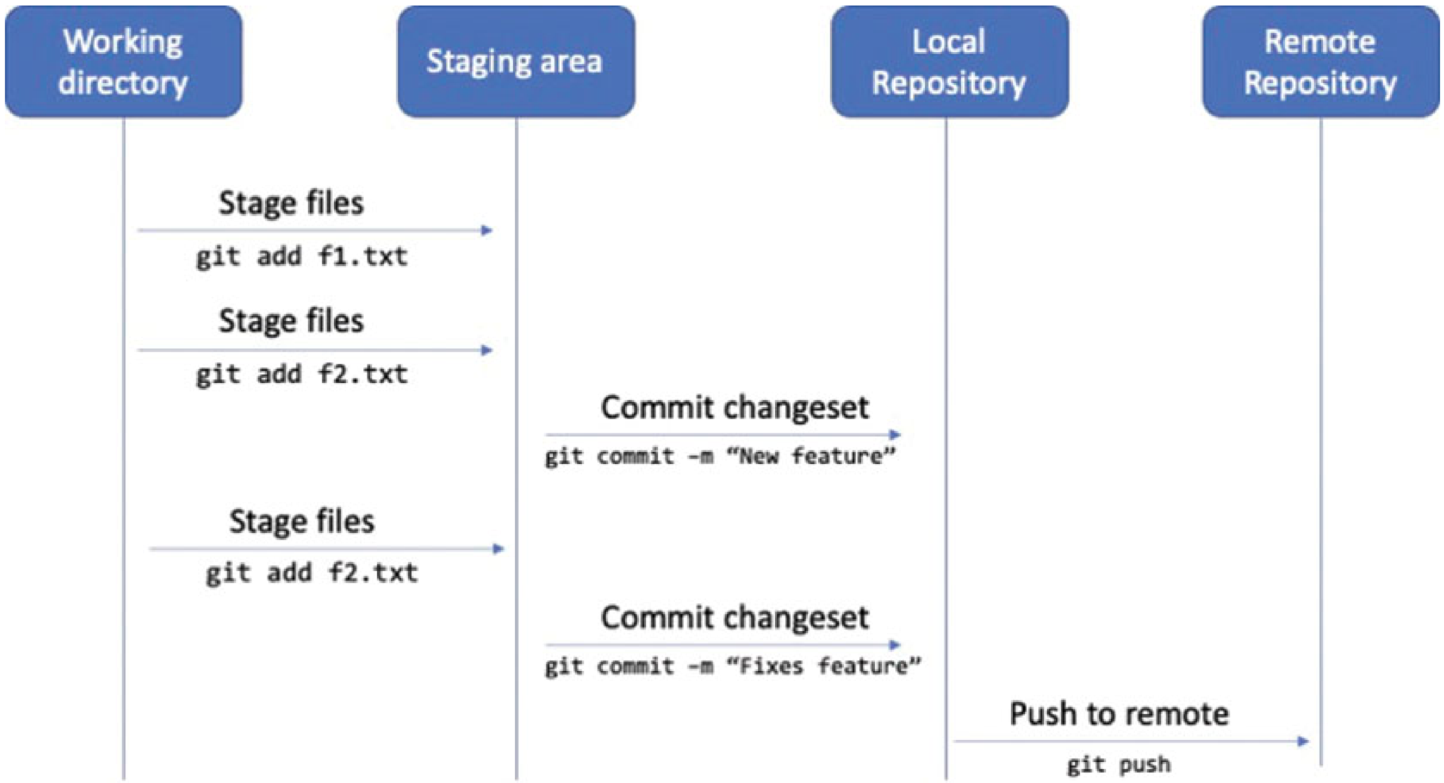


Figure A-4 Commit process in Git

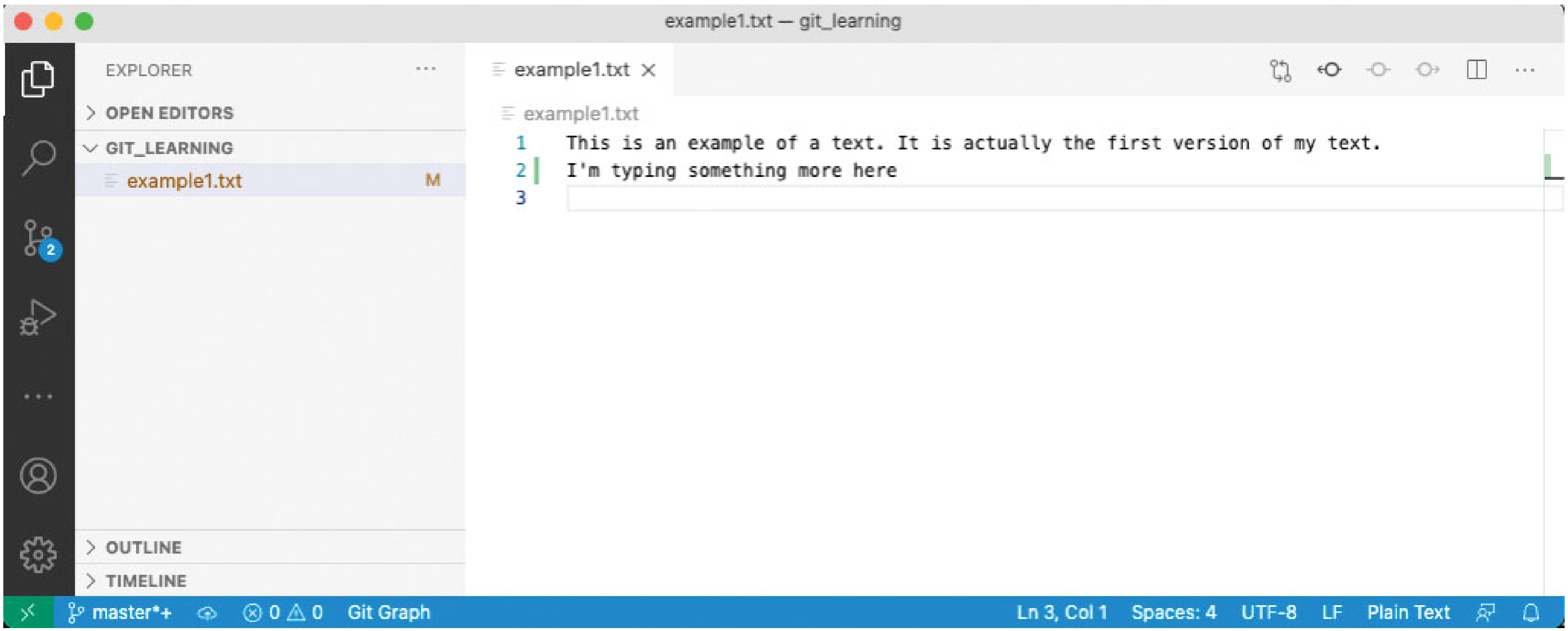


Figure A-5 Updating the first file

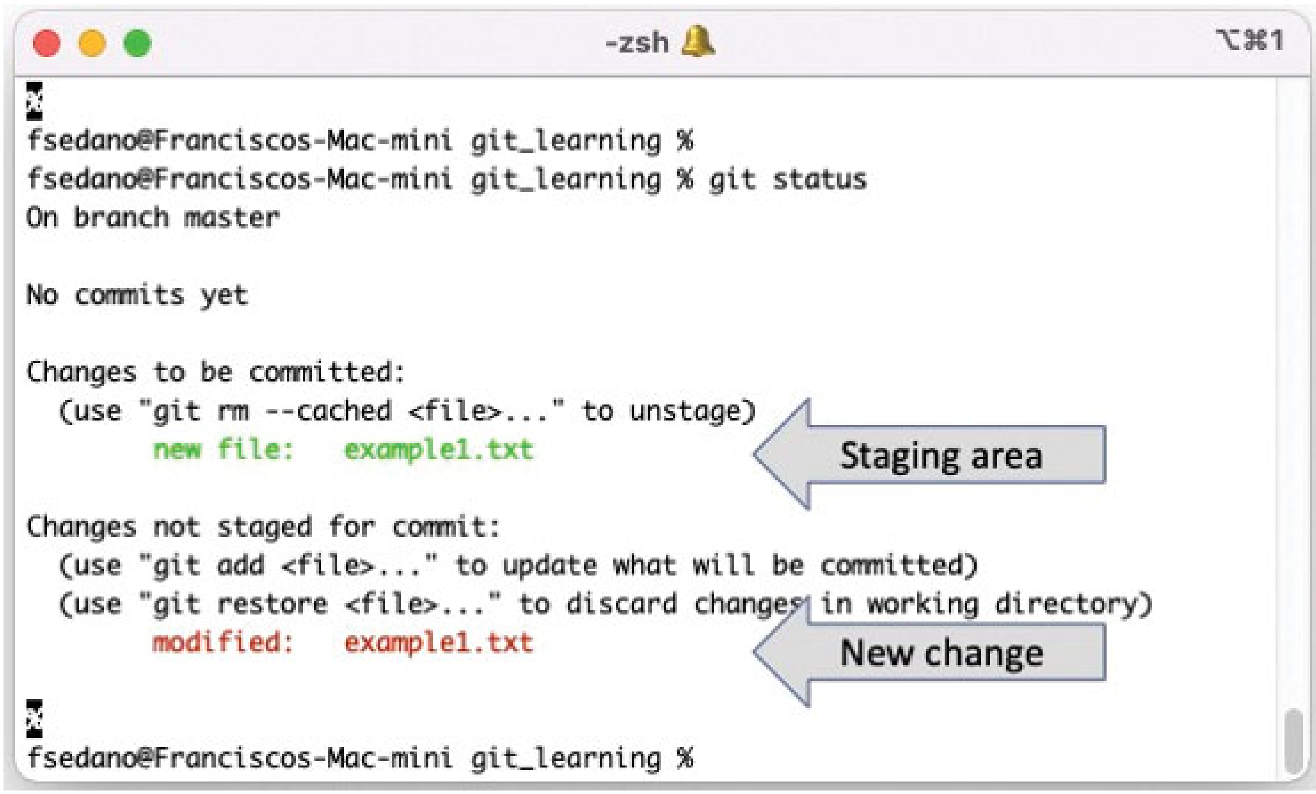


Figure A-6 Git detecting changed files

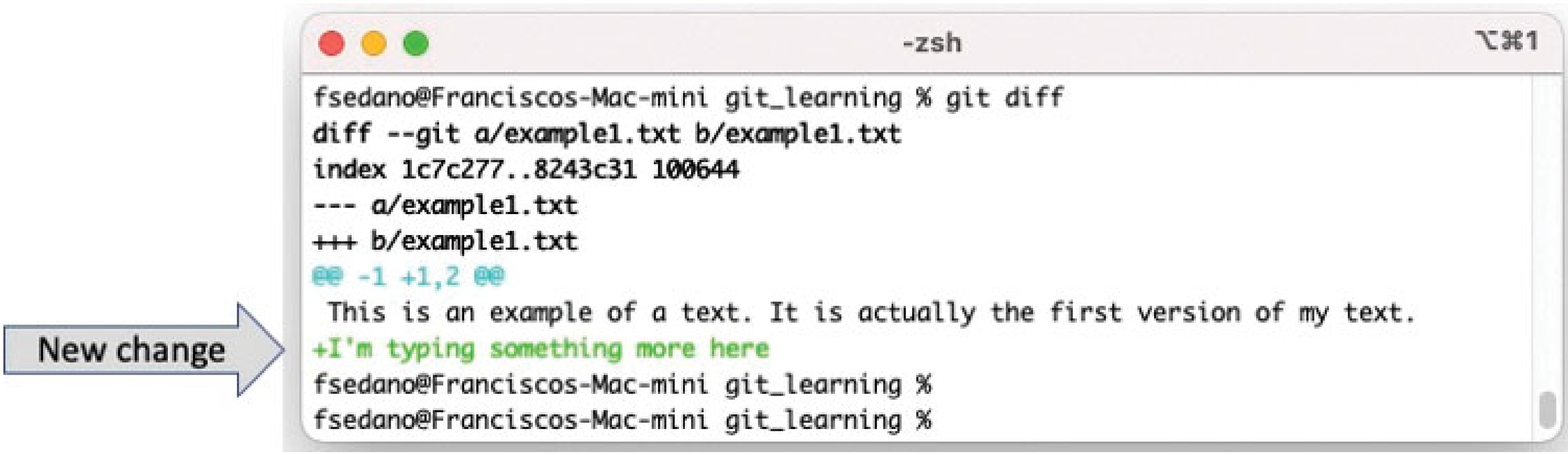


Figure A-7 Diff between two file versions

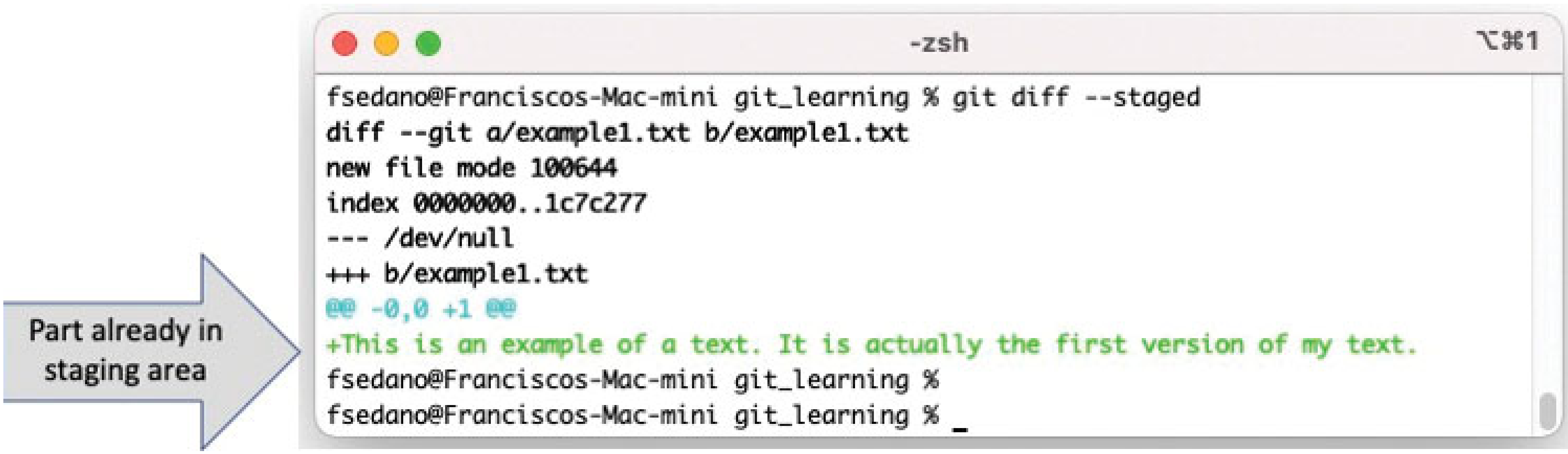


Figure A-8 Git displaying staged files

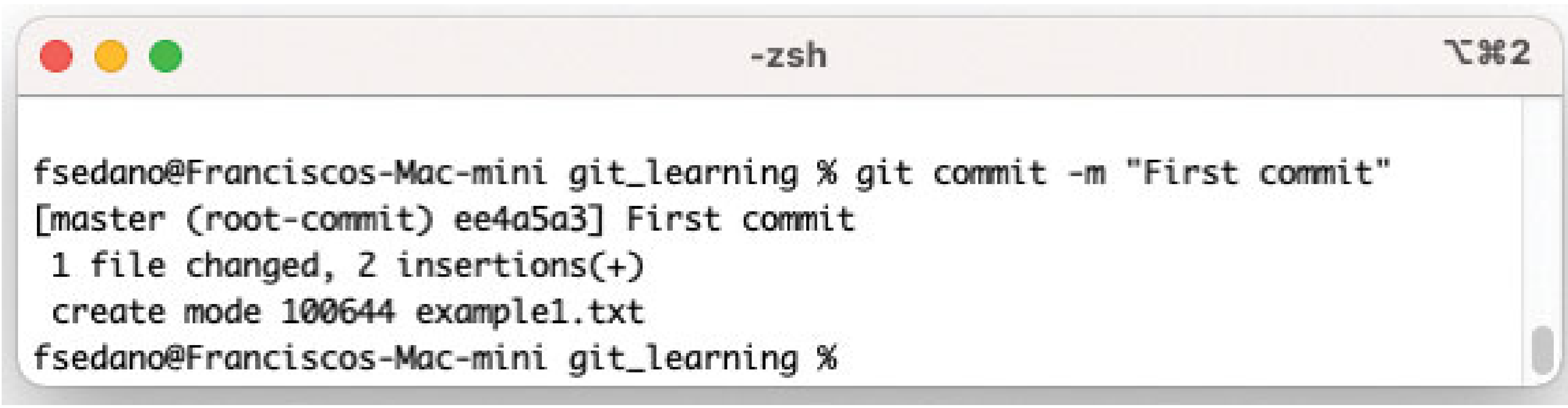


Figure A-9 Committing changes

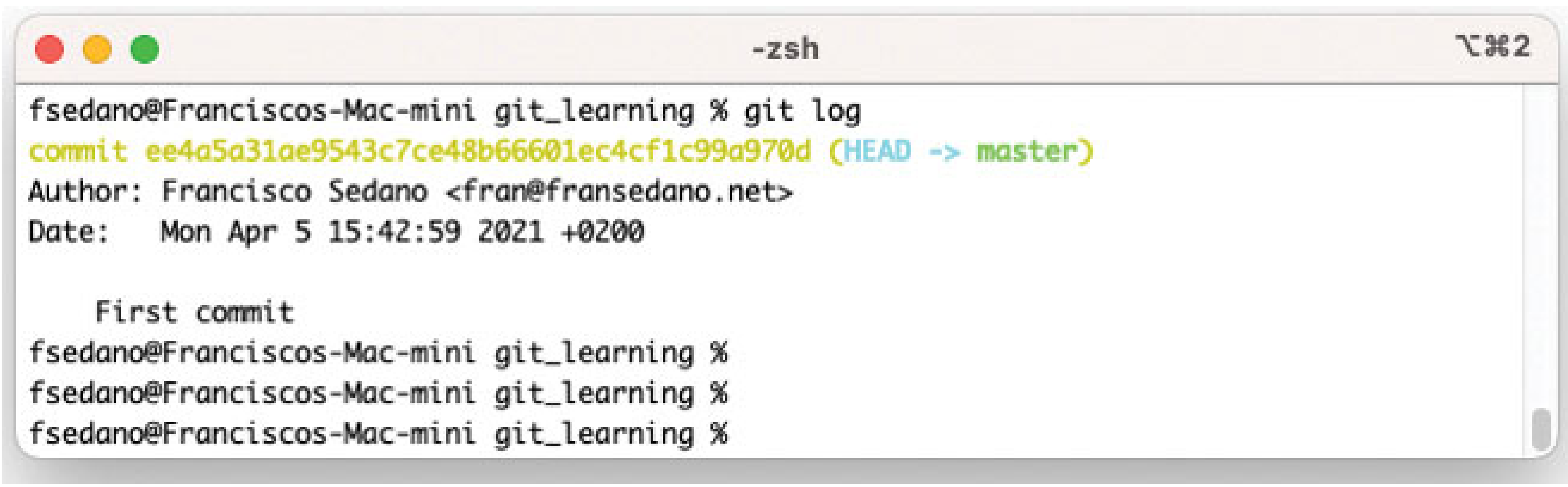


Figure A-10 Examining Git repository changes

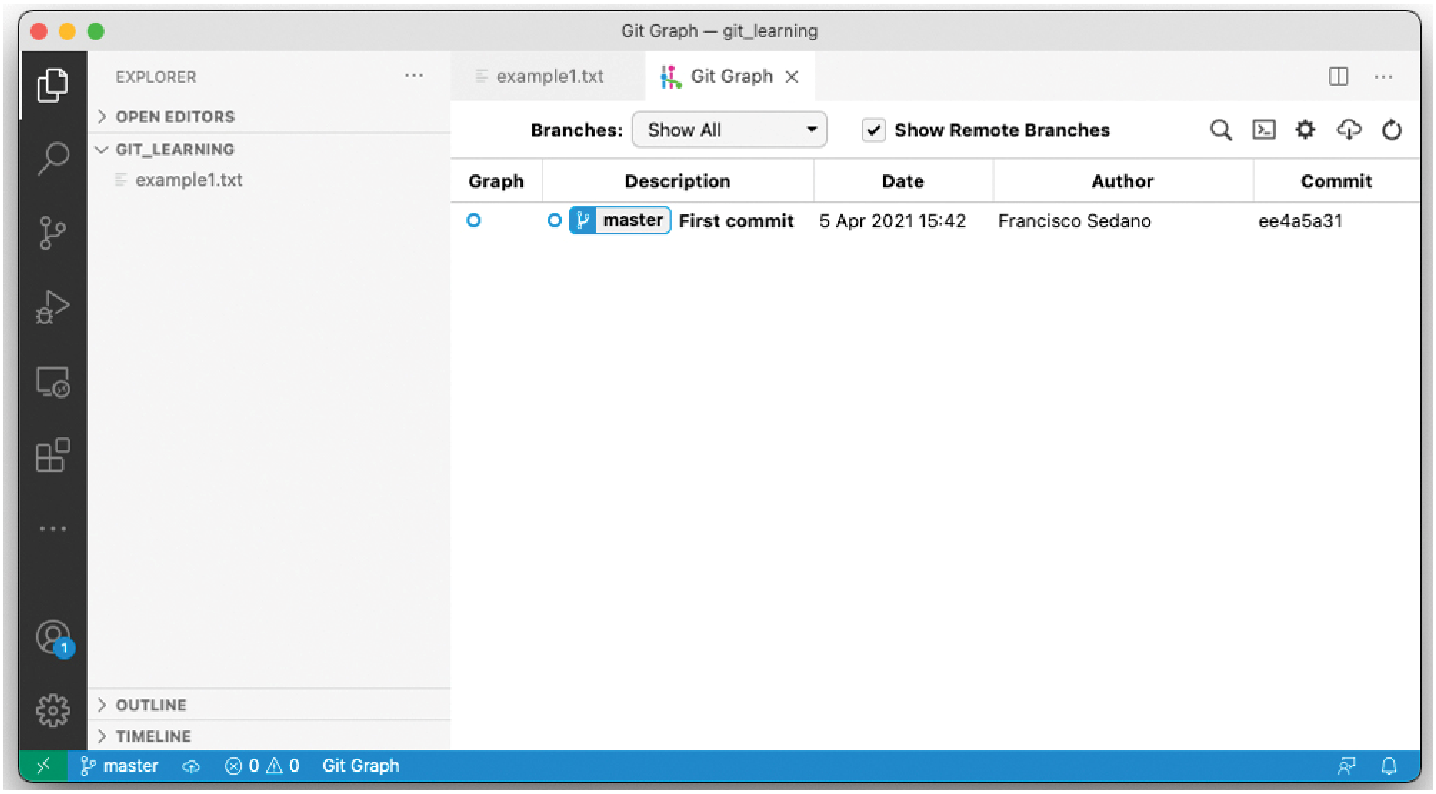


Figure A-11 Viewing repository activity

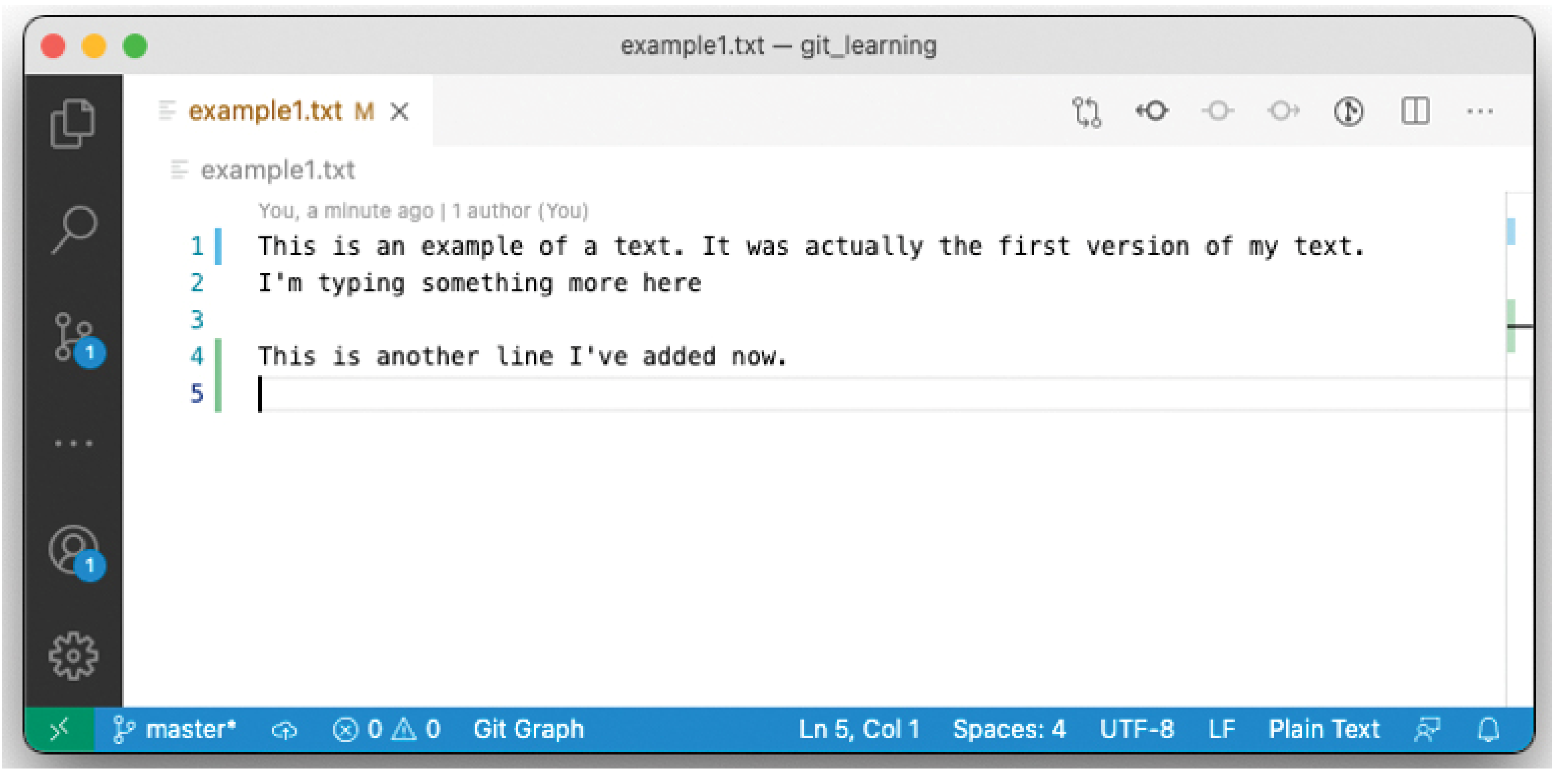


Figure A-12 Marks in Visual Studio Code

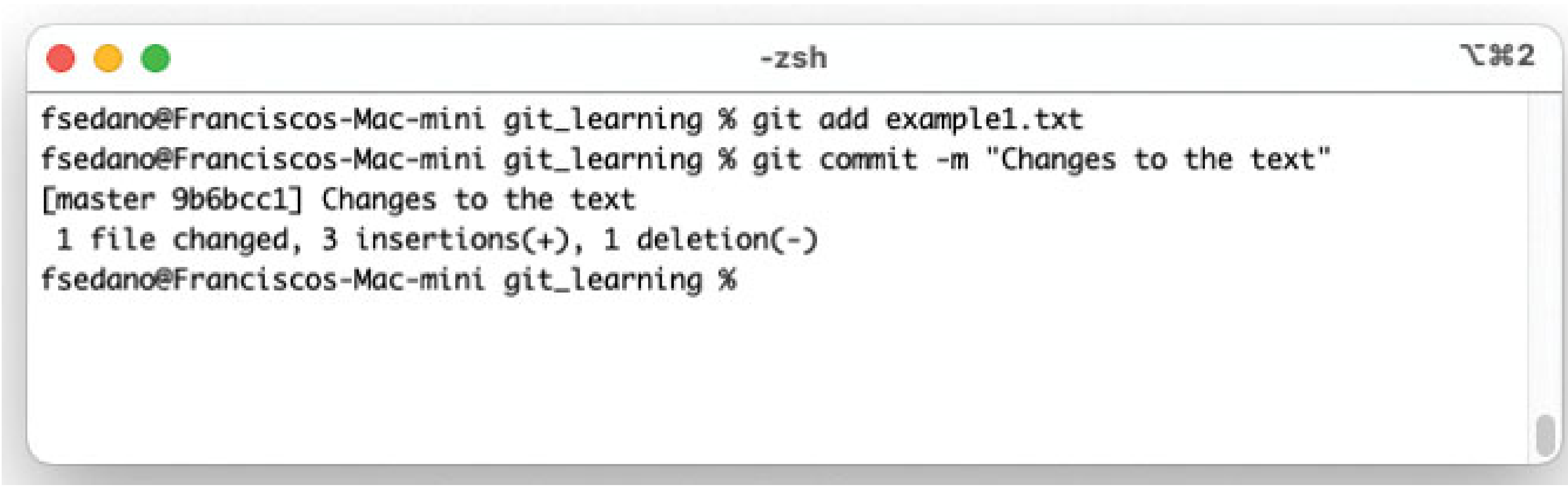


Figure A-13 Committing changes

Figure A-14 Viewing commit history with git log

```
-zsh                                                        ⌥⌘2
fsedano@Franciscos-Mac-mini git_learning % git show 9b6bcc19e95fbe99c89d68339ddb0f6fbcfb1b1f
commit 9b6bcc19e95fbe99c89d68339ddb0f6fbcfb1b1f (HEAD -> master)
Author: Francisco Sedano <fran@fransedano.net>
Date:   Mon Apr 5 15:55:09 2021 +0200

    Changes to the text

diff --git a/example1.txt b/example1.txt
index 8243c31..393ffb0 100644
--- a/example1.txt
+++ b/example1.txt
@@ -1,2 +1,4 @@
-This is an example of a text. It is actually the first version of my text.
+This is an example of a text. It was actually the first version of my text.
 I'm typing something more here
+
+This is another line I've added now.
fsedano@Franciscos-Mac-mini git_learning %
fsedano@Franciscos-Mac-mini git_learning %
```

Figure A-15 Displaying changes in a commit using the command line

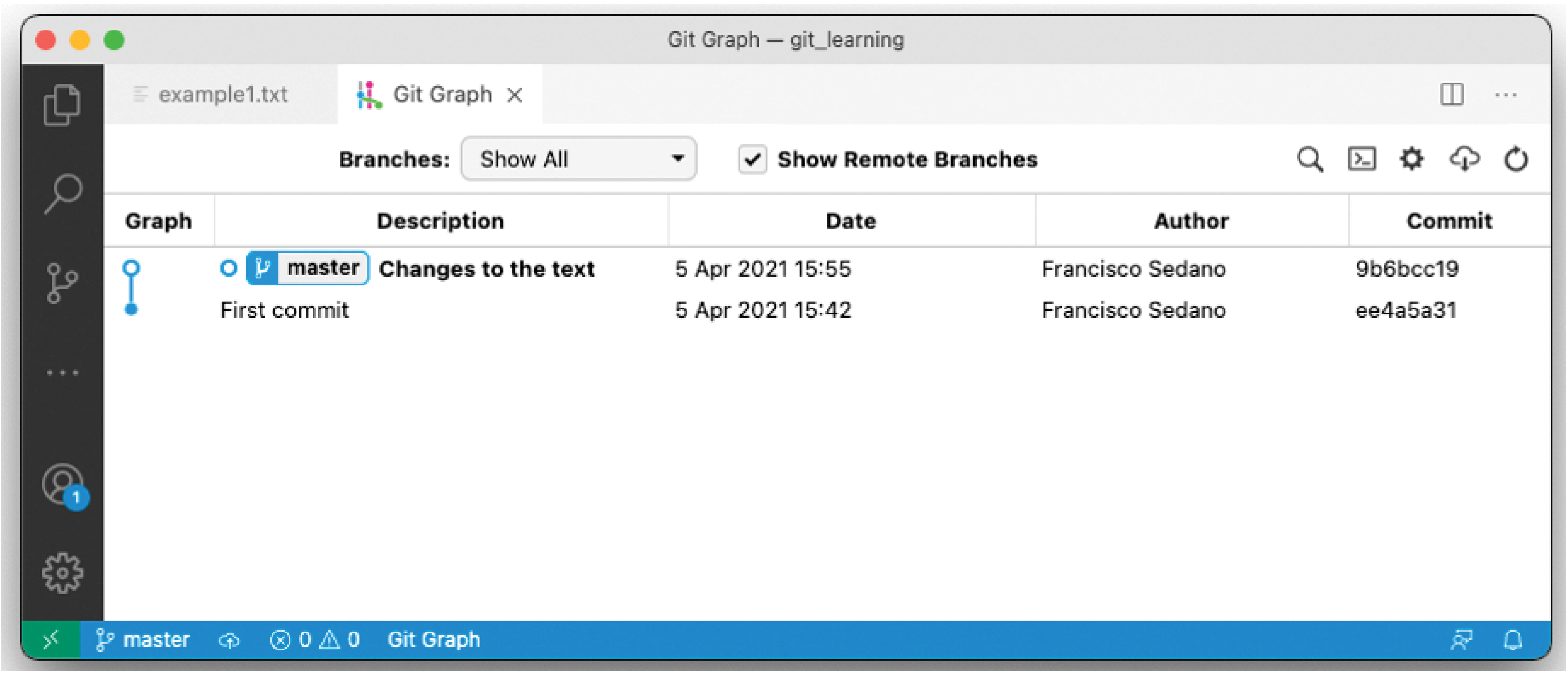


Figure A-16 Displaying commits using Visual Studio Code

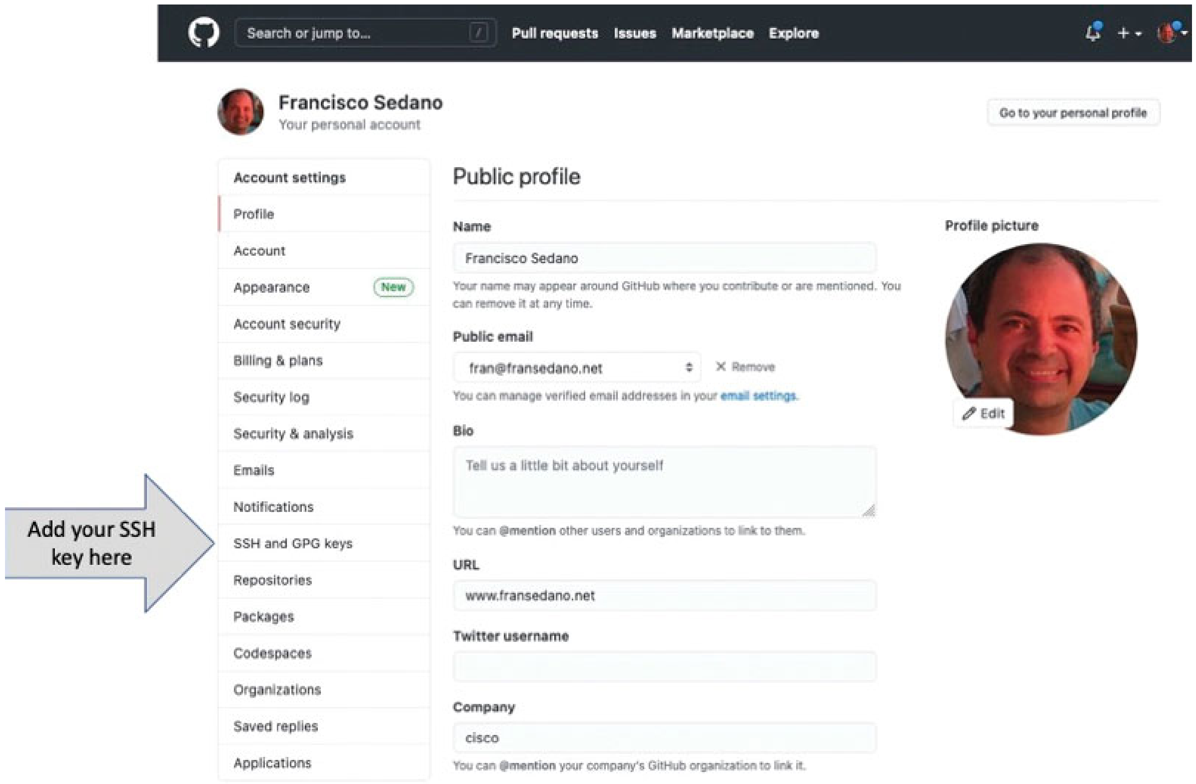


Figure A-17 Adding SSH keys to GitHub

From *Understanding and Troubleshooting Cisco Catalyst 9800 Series Wireless Controller by* Simone Arena, Francisco Sedano Crippa, Nicolas Darchis and Sudha Katgeri(9780137492329) Cisco Press

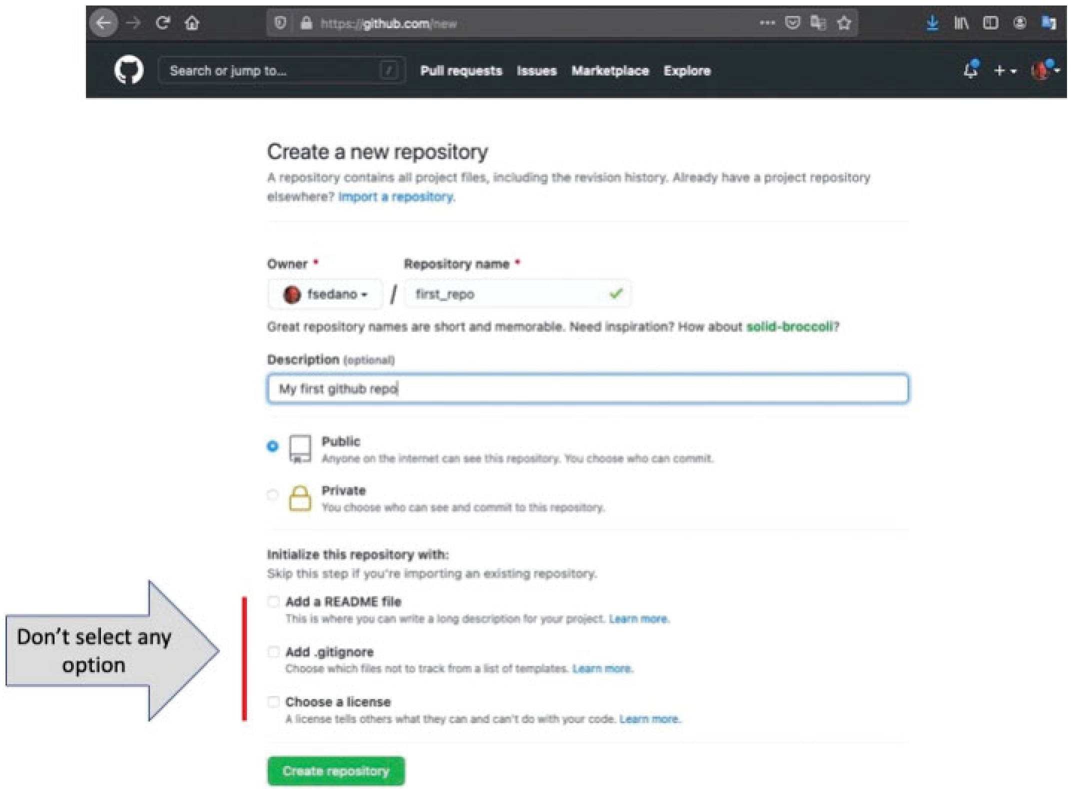


Figure A-18 Creating a new repository in GitHub

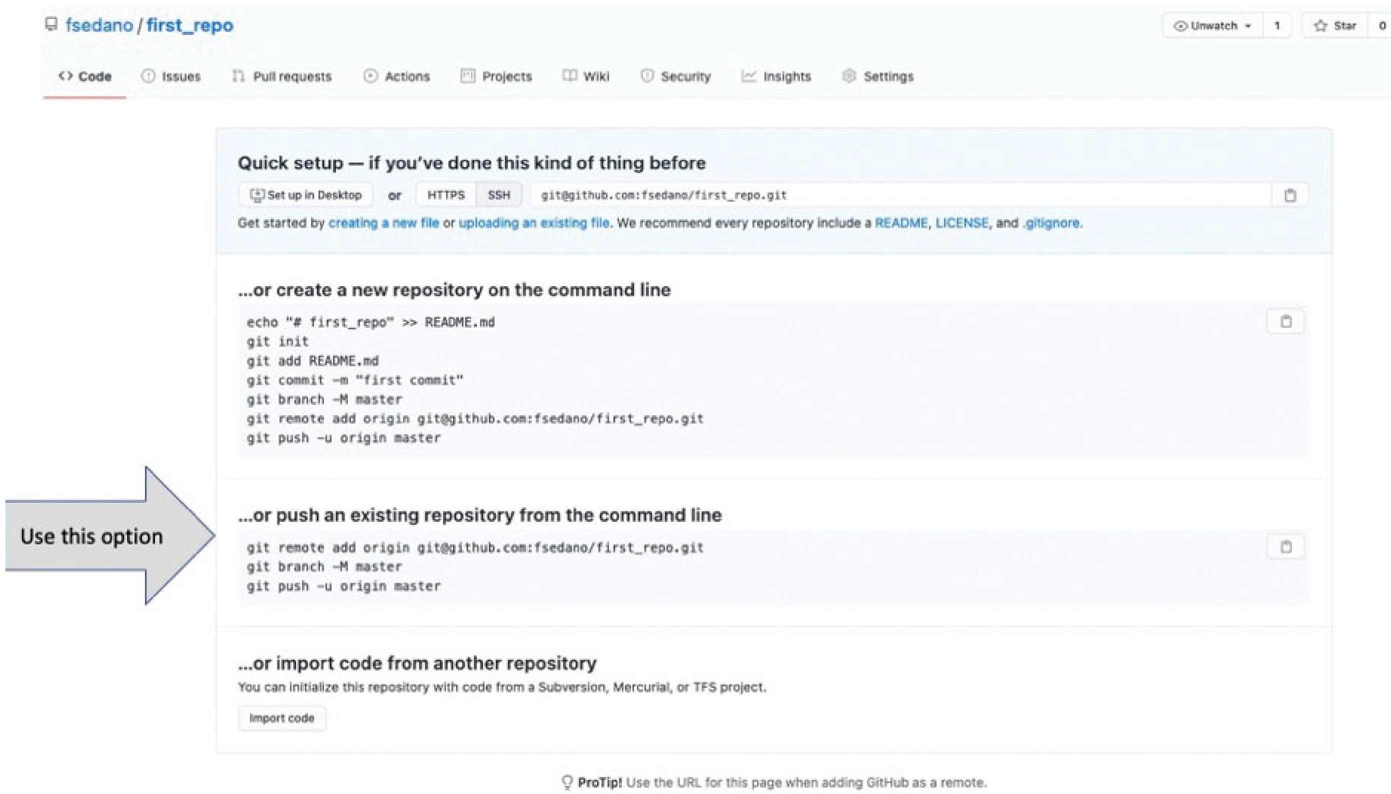


Figure A-19 New repository in GitHub

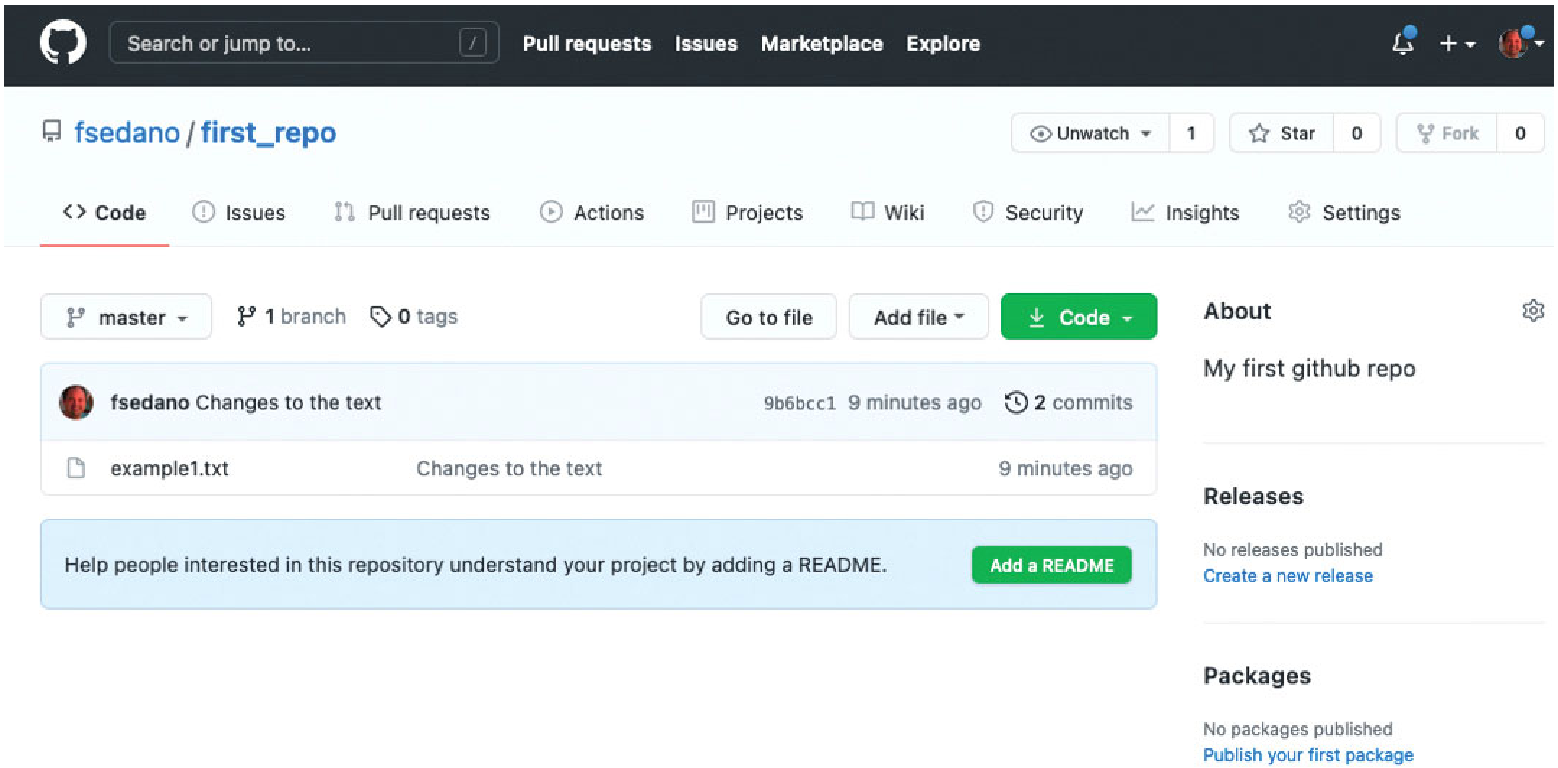


Figure A-20 Repository after the first commit

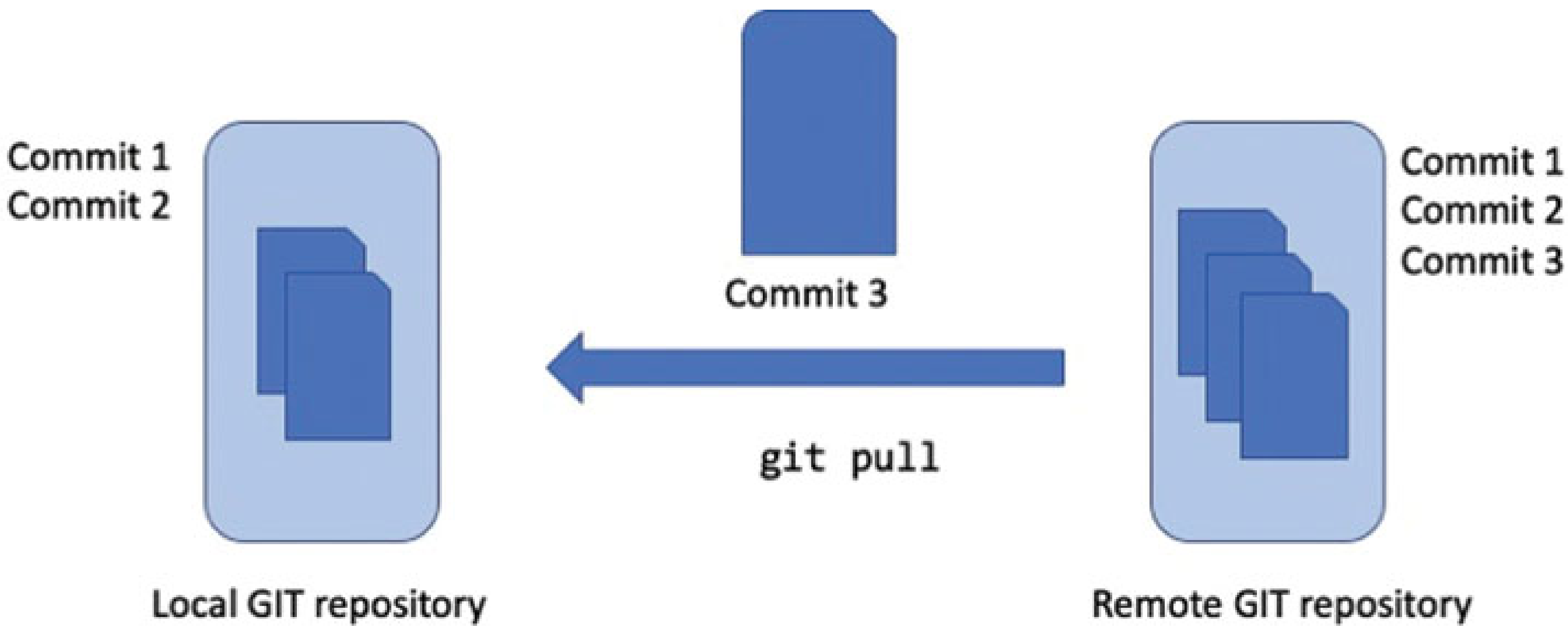


Figure A-21 Pulling from remote repositories

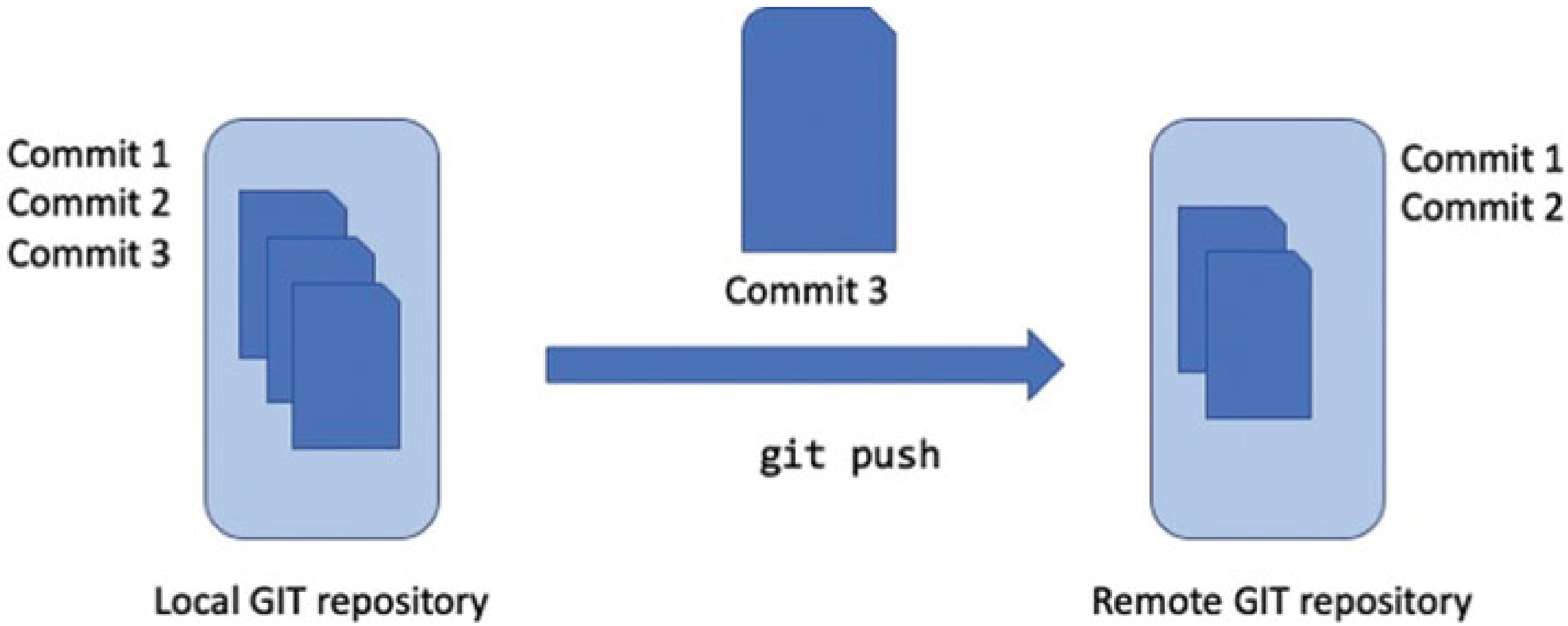
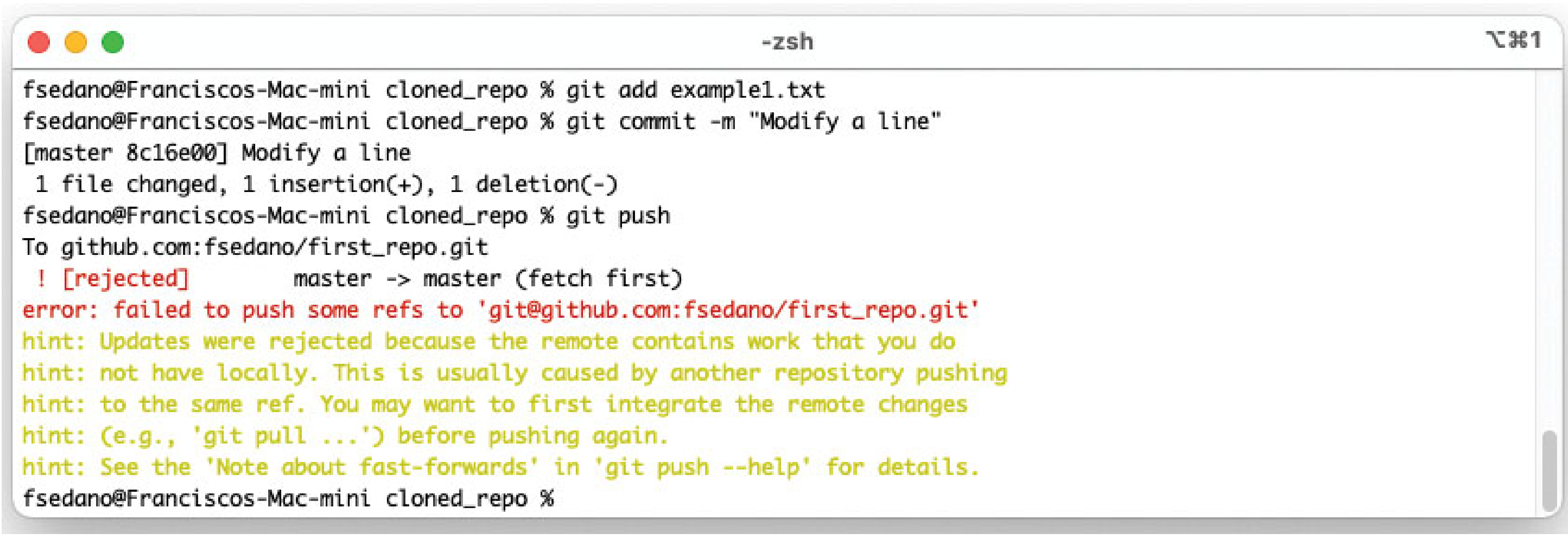


Figure A-22 Pushing to remote repositories



```
-zsh                                                        ⌥⌘1
fsedano@Franciscos-Mac-mini cloned_repo % git add example1.txt
fsedano@Franciscos-Mac-mini cloned_repo % git commit -m "Modify a line"
[master 8c16e00] Modify a line
 1 file changed, 1 insertion(+), 1 deletion(-)
fsedano@Franciscos-Mac-mini cloned_repo % git push
To github.com:fsedano/first_repo.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'git@github.com:fsedano/first_repo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
fsedano@Franciscos-Mac-mini cloned_repo %
```

Figure A-23 Git message if the local repository lags behind a remote one

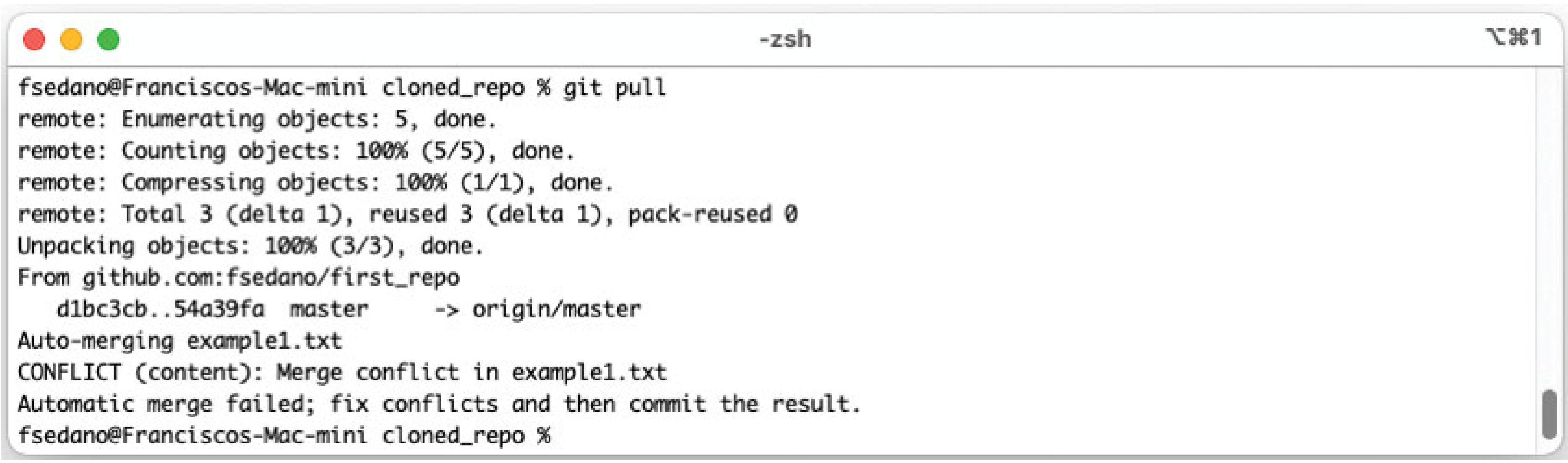


Figure A-24 Git conflicts

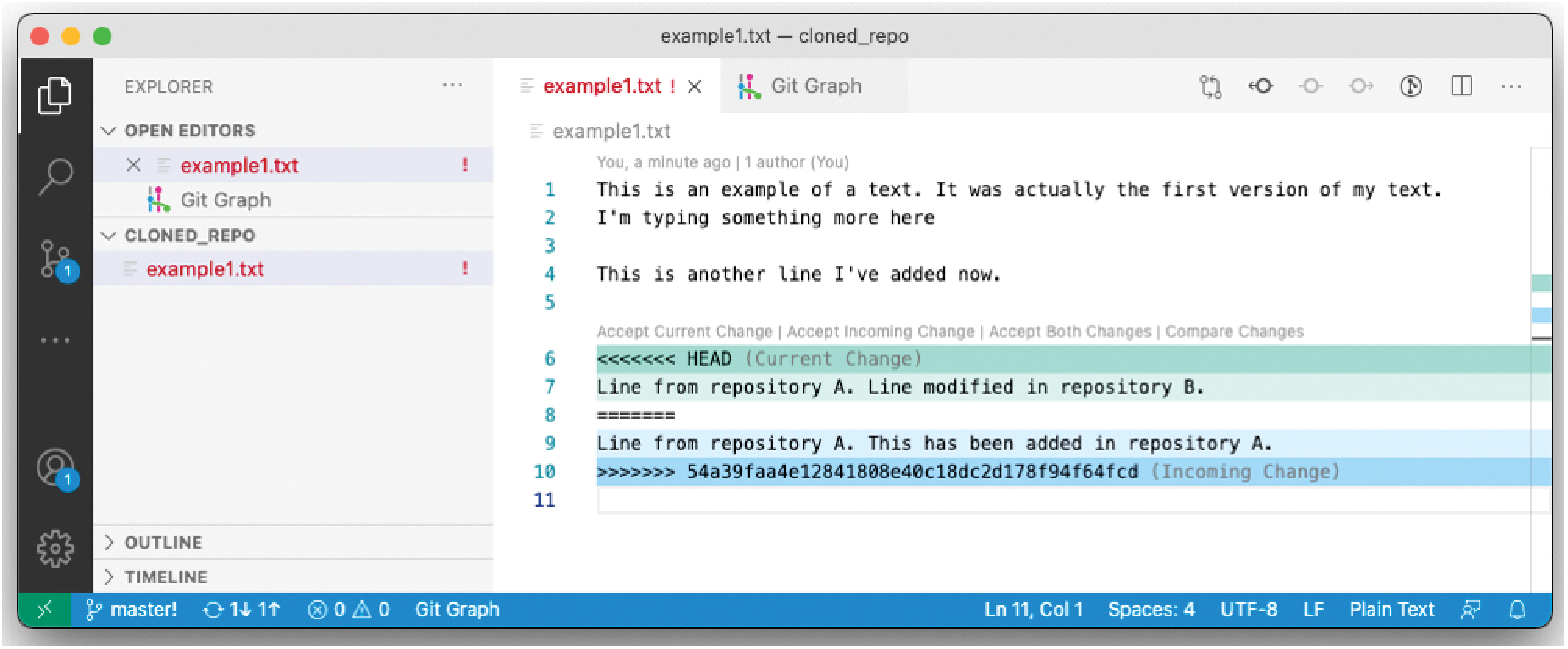


Figure A-25 Solving Git conflicts

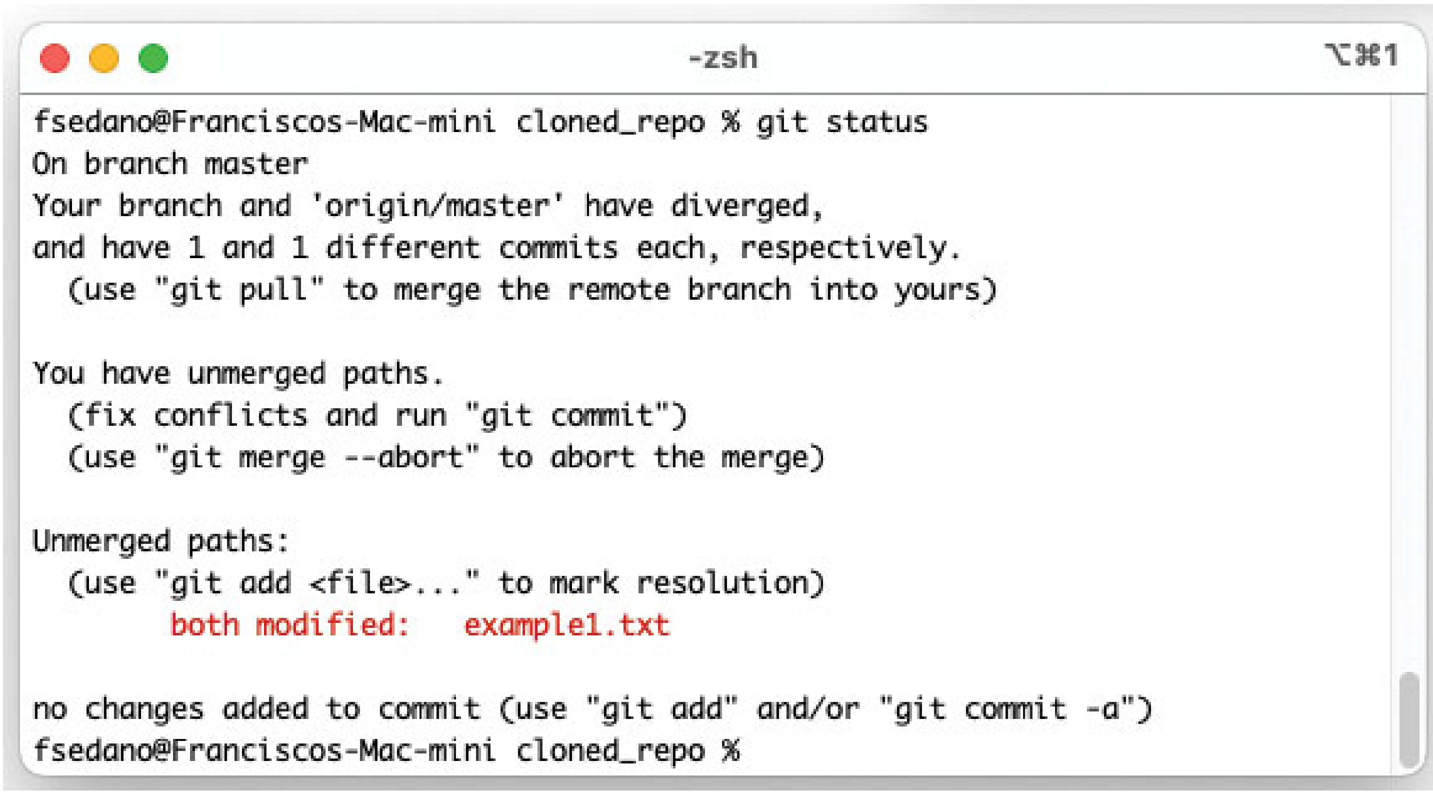


Figure A-26 Solving Git conflicts

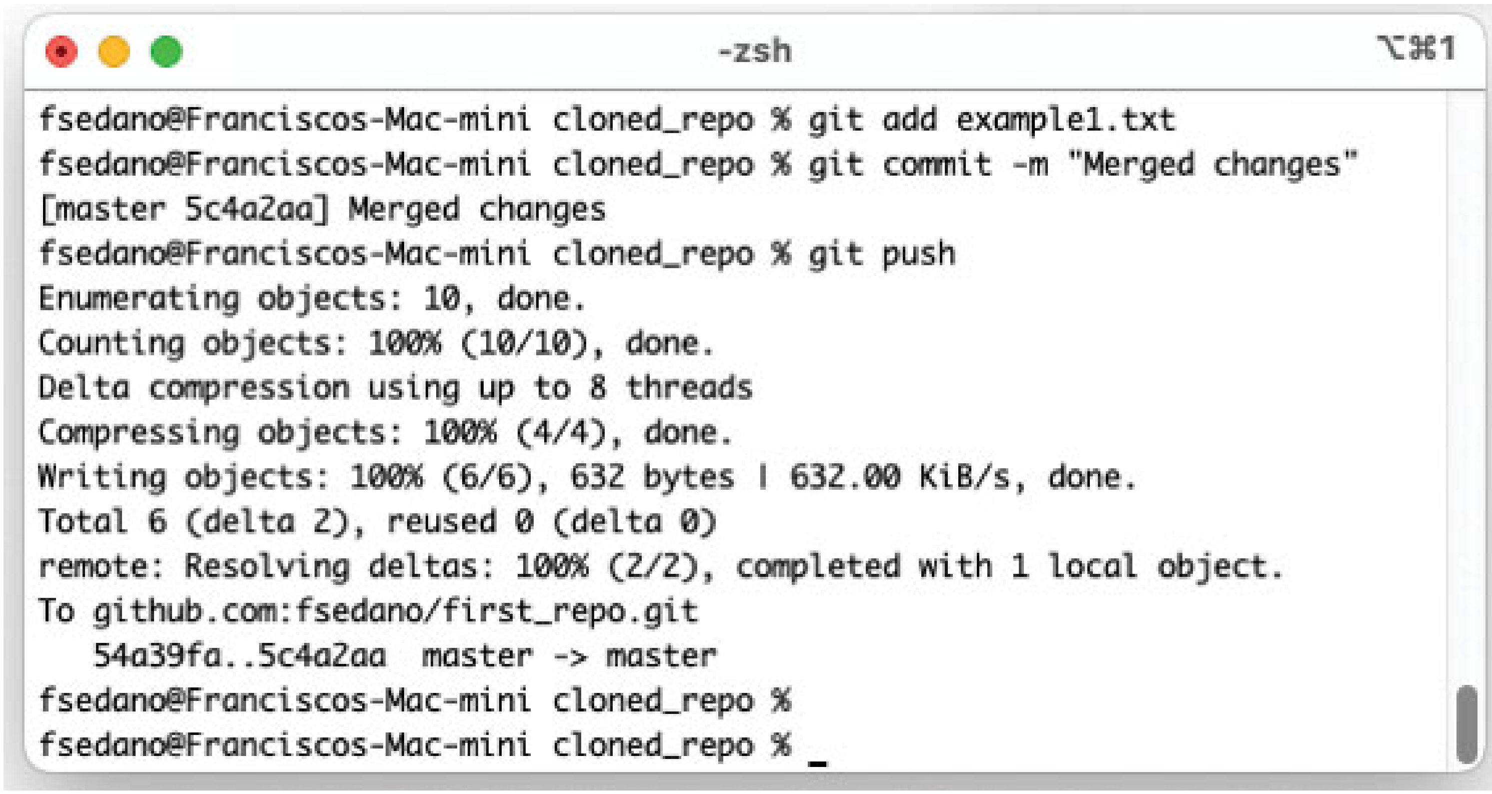


Figure A-27 Git conflict solved

From *Understanding and Troubleshooting Cisco Catalyst 9800 Series Wireless Controller by* Simone Arena, Francisco Sedano Crippa, Nicolas Darchis and Sudha Katgeri(9780137492329) Cisco Press

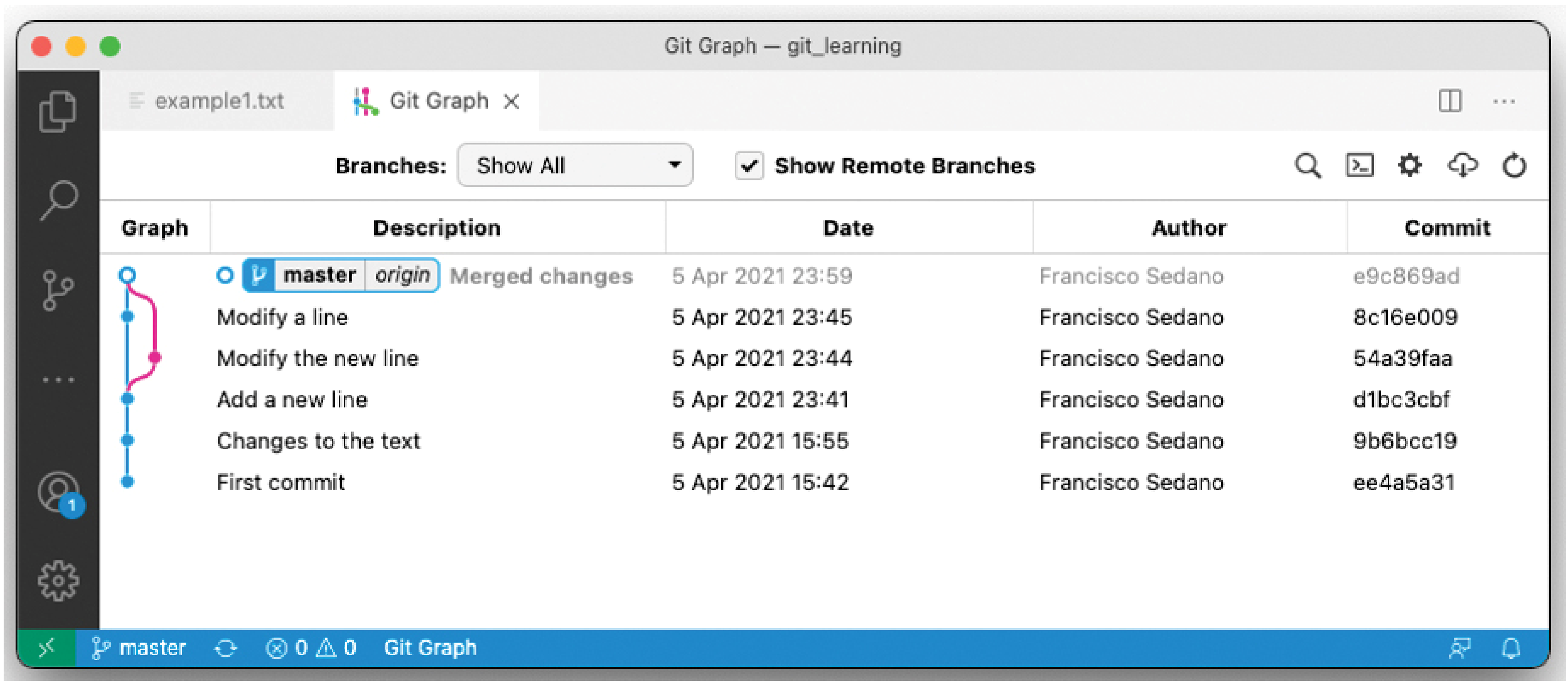


Figure A-28 Git history flow

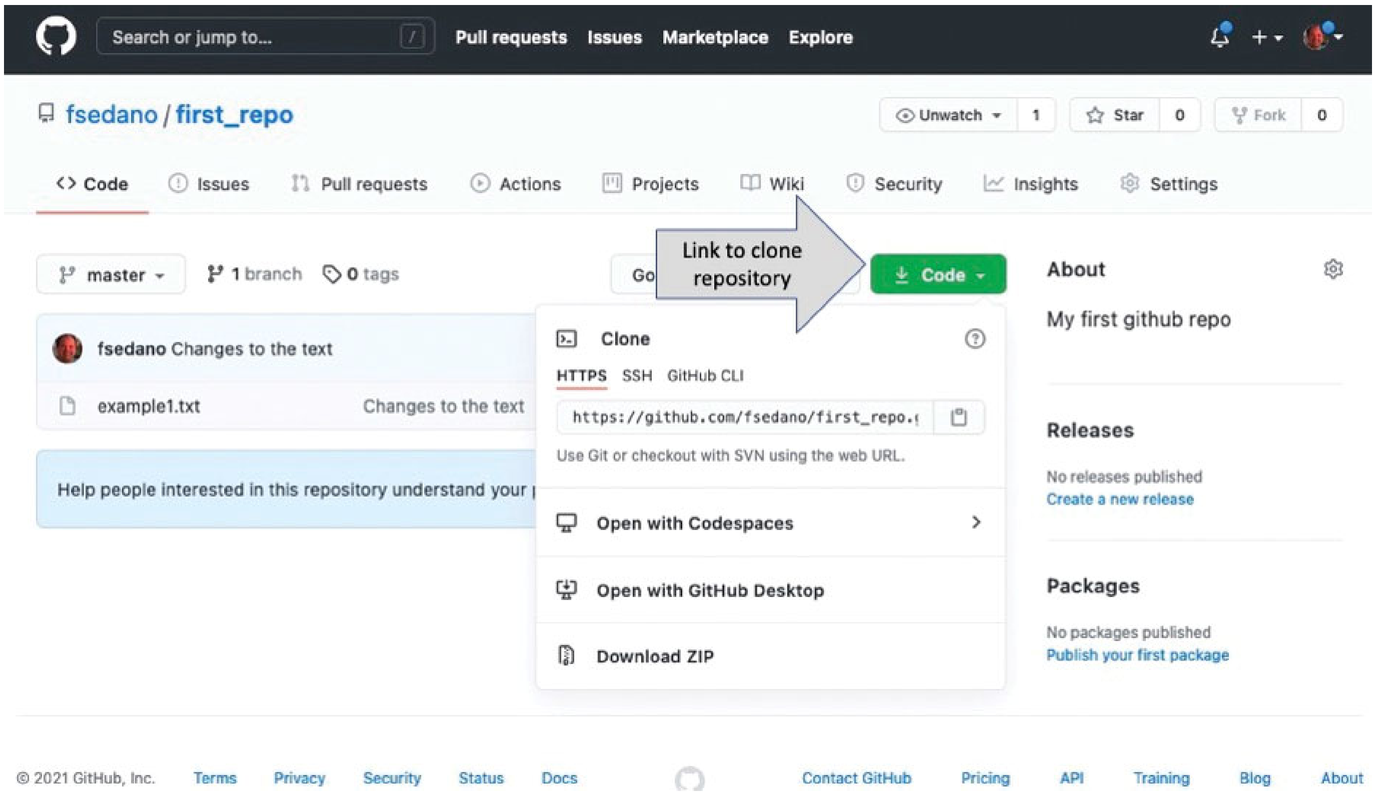


Figure A-29 Sharing repositories using GitHub

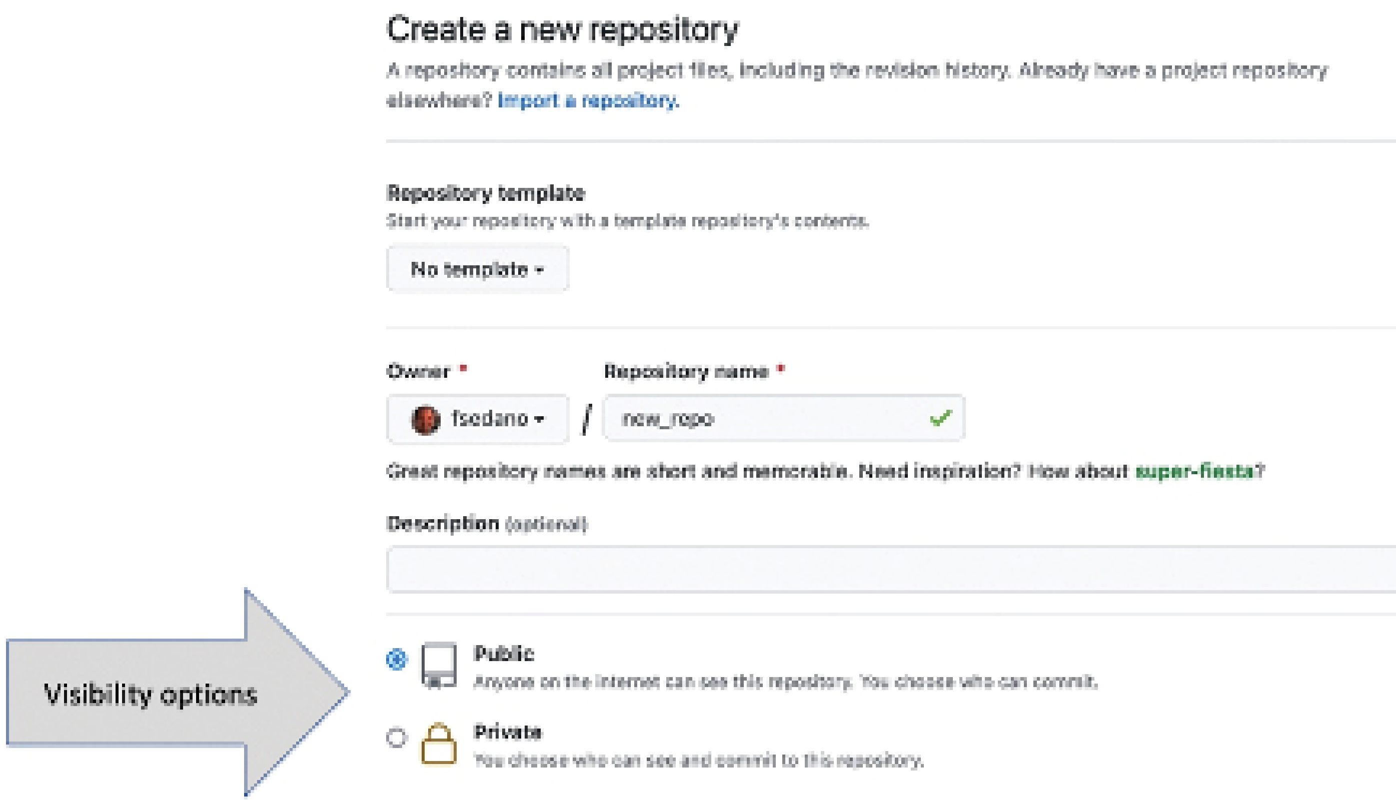


Figure A-30 Cloning Git repositories

```
fsedano@Franciscos-Mac-mini ~ % git clone https://github.com/fsedano/first_repo.git
Cloning into 'first_repo'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
fsedano@Franciscos-Mac-mini ~ % cd first_repo
fsedano@Franciscos-Mac-mini first_repo % ls
example1.txt
fsedano@Franciscos-Mac-mini first_repo % git log
commit 9b6bcc19e95fbe99c89d68339ddb0f6fbcfb1b1f (HEAD -> master, origin/master, origin/HEAD)
Author: Francisco Sedano <fran@fransedano.net>
Date:   Mon Apr 5 15:55:09 2021 +0200

    Changes to the text

commit ee4a5a31ae9543c7ce48b66601ec4cf1c99a970d
Author: Francisco Sedano <fran@fransedano.net>
Date:   Mon Apr 5 15:42:59 2021 +0200

    First commit
fsedano@Franciscos-Mac-mini first_repo %
```

Figure A-31 Choosing repository visibility at creation time

Figure A-32 Creating a Python virtual environment

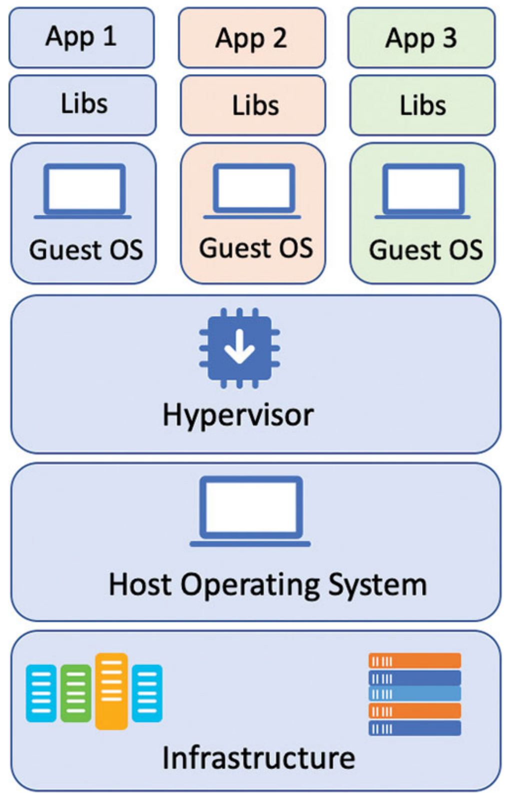


Figure A-33 Hypervisor architecture

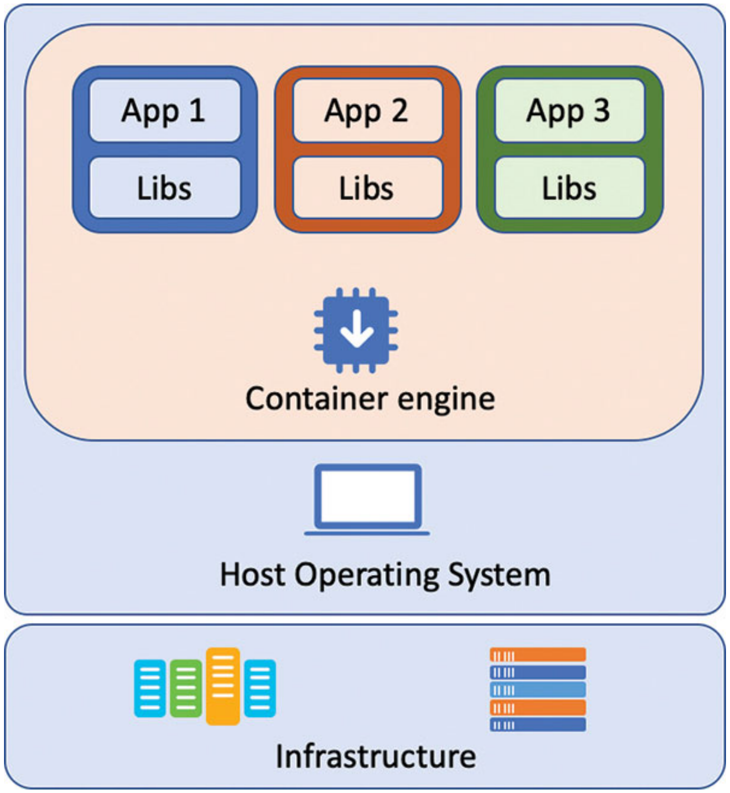


Figure A-34 Container architecture

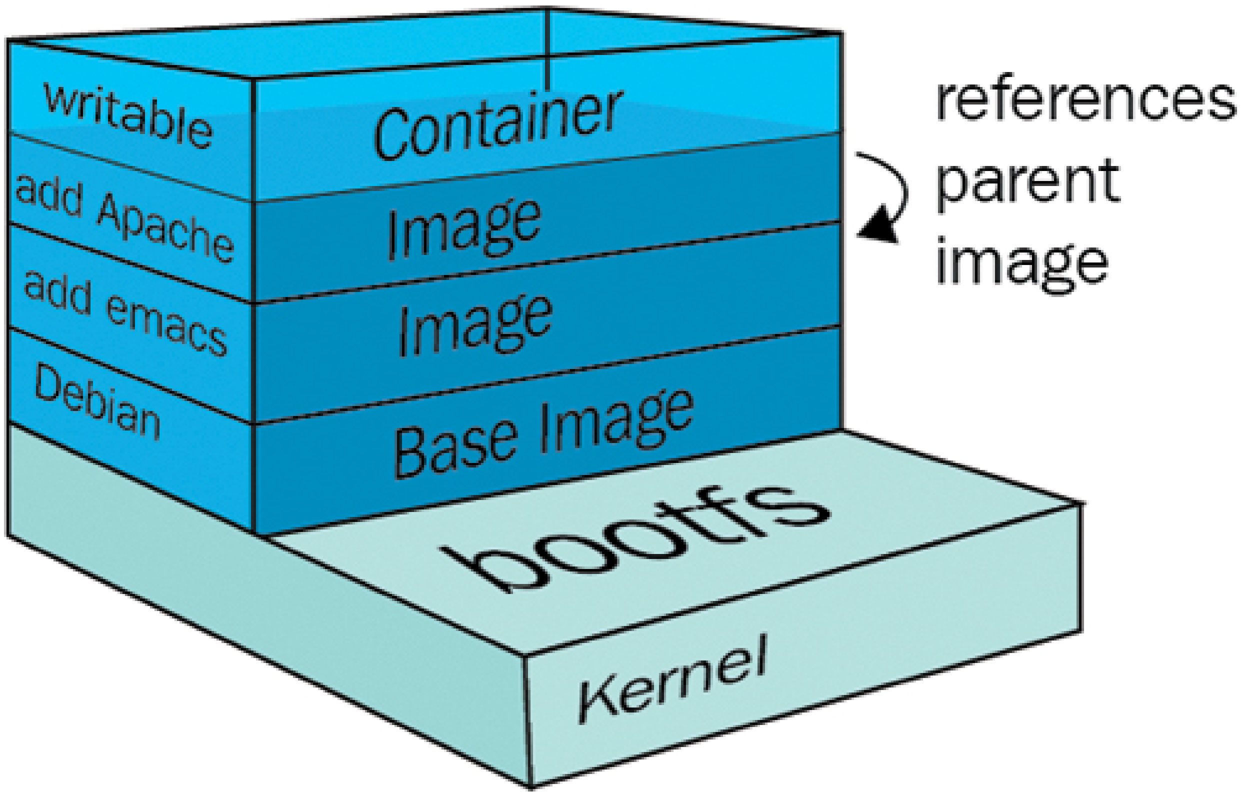


Figure A-35 Layers in a container image

Figure A-36 Layers created by a Dockerfile

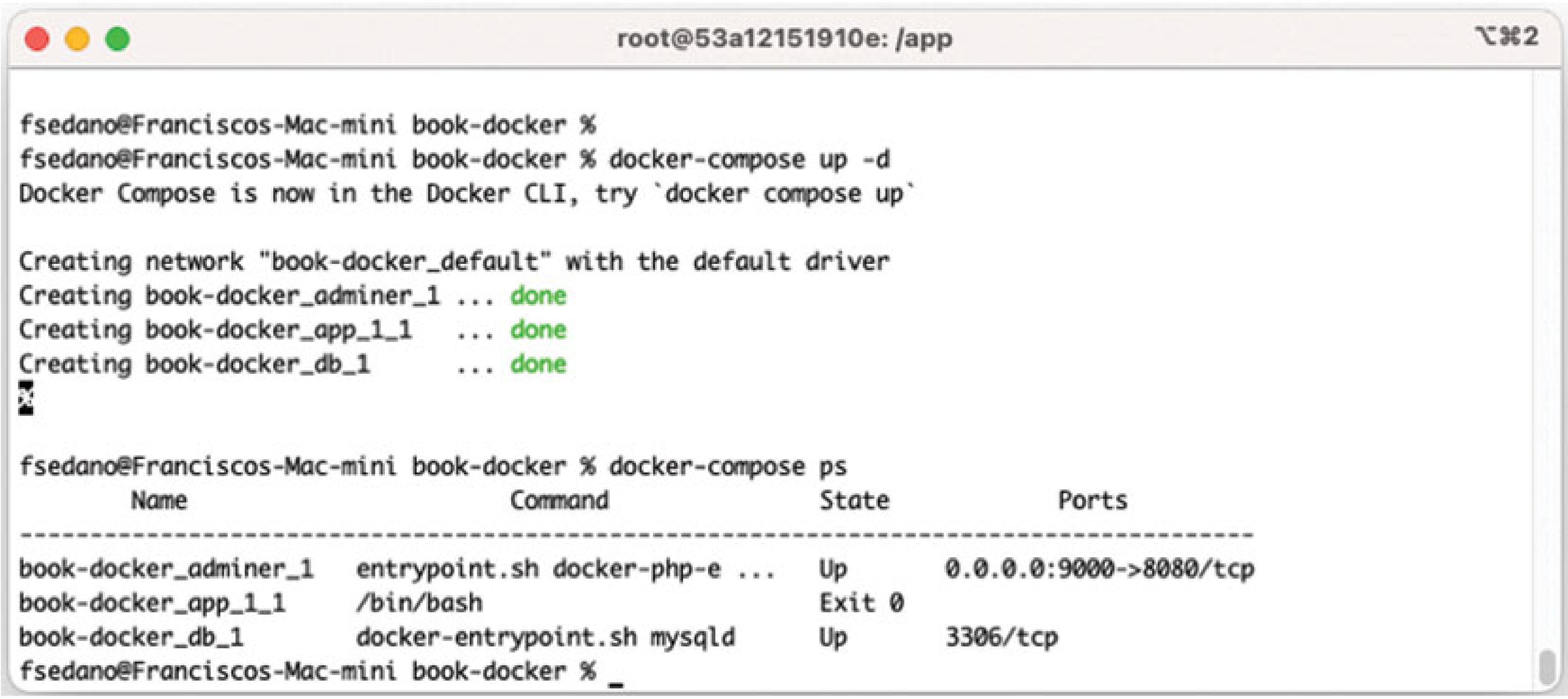


Figure A-37 Checking container status by using docker-compose