# Cloud Native BNG Control Plane Configuration Guide, Release 2021.01.0

**First Published:** 2021-02-26

**Americas Headquarters**
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

# CONTENTS

APPENDIX A

# About this Guide

**Note** The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. While any existing biased terms are being substituted, exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

This preface describes the Cloud Native Broadband Network Gateway (cnBNG) Control Plane (CP) Configutration Guide, how it is organized, and its document conventions.

This guide describes the Cloud Native BNG solution and includes feature descriptions, specification compliance, session flows, configuration instructions, CLI commands and so on.

• Conventions Used, on page xi

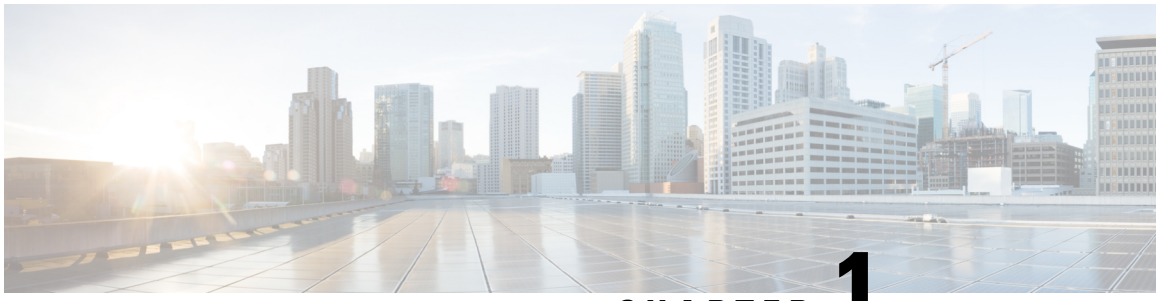
# Conventions Used

The following tables describe the conventions used throughout this documentation.

| Notice Type | Description |
|---|---|
| Information Note | Provides information about important features or instructions. |
| Caution | Alerts you of potential damage to a program, device, or system. |
| Warning | Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards. |

| Typeface Conventions | Description |
|---|---|
| Text represented as a `screen display` | This typeface represents displays that appear on your terminal screen, for example:<br>`Login:` |

| Typeface Conventions | Description |
|---|---|
| Text represented as **commands** | This typeface represents commands that you enter, for example:<br>**show ip access-list**<br>This document always gives the full form of a command in lowercase letters. Commands are not case sensitive. |
| Text represented as a **command** *variable* | This typeface represents a variable that is part of a command, for example:<br>**show card** *slot_number*<br>*slot_number* is a variable representing the desired chassis slot number. |
| Text represented as menu or sub-menu names | This typeface represents menus and sub-menus that you access within a software application, for example:<br>Click the **File** menu, then click **New** |



CHAPTER 1

# cnBNG Overview



# Feature Summary and Revision History

## Summary Data

***Table 1: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | Cloud Native Broadband Network Gateway |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Not Applicable |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

# Overview

This chapter provides an overview of the Cloud Native Broadband Network Gateway (cnBNG) solution.

The Broadband Network Gateway (BNG) is the access point for subscribers, through which they connect to the broadband network. When a connection is established between BNG and Customer Premise Equipment (CPE), the subscriber can access the broadband services provided by the Network Service Provider (NSP) or Internet Service Provider (ISP).

BNG establishes and manages subscriber sessions. When a session is active, BNG aggregates traffic from various subscriber sessions from an access network, and routes it to the network of the service provider.

BNG is deployed by the service provider and is present at the first aggregation point in the network, such as the edge router. An edge router, like the Cisco ASR 9000 Series Router, needs to be configured to act as the BNG. Because the subscriber directly connects to the edge router, BNG effectively manages subscriber access, and subscriber management functions such as:

- Authentication, Authorization, and Accounting (AAA) of subscriber sessions
- Address assignment
- Security
- Policy management
- Quality of Service (QoS)

Implementing the BNG provides the following benefits:

- Communicates with authentication, authorization, and accounting (AAA) server to perform session management and billing functions besides the routing function. This feature makes the BNG solution more comprehensive.
- Provides different network services to the subscriber. This enables the service provider to customize the broadband package for each customer based on their needs.

Cisco provides two BNG solutions:

- **Physical BNG** where the BNG Control Plane (CP) and the User Plane (UP) are tightly coupled inside a Cisco IOS XR platform where the CP runs on an x86 CPU and the UP runs on a physical NPU or ASIC.

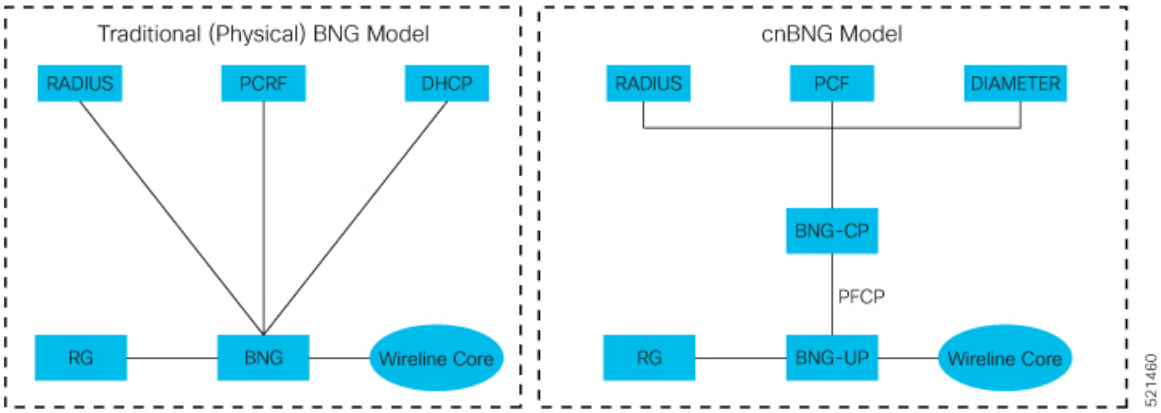
  For more information about the physical BNG, refer to the latest version of the *Broadband Network Gateway Configuration Guide* for Cisco ASR 9000 Series Routers.

- **Virtual BNG (vBNG)** where the BNG CP and UP run in separate VM-based Cisco IOS XR software on general purpose x86 UCS servers.

# Evolution of cnBNG

The Cisco Cloud Native Broadband Network Gateway (cnBNG) provides a new dimension to the Control Plane and User Plane Separation (CUPS) architecture of the Broadband Network Gateway (BNG), enabling flexibility and rapid scaling for Internet Service Providers (ISPs).

*Figure 1: Evolution of BNG to cnBNG*




The architectural change is an evolution from an integrated traditional BNG running on a single router to a disaggregated solution, where the centralized subscriber management runs on an elastic and scalable Cloud Native Control Plane (CP) and the User Plane (UP) delivers the forwarding functionality.

# cnBNG Architecture

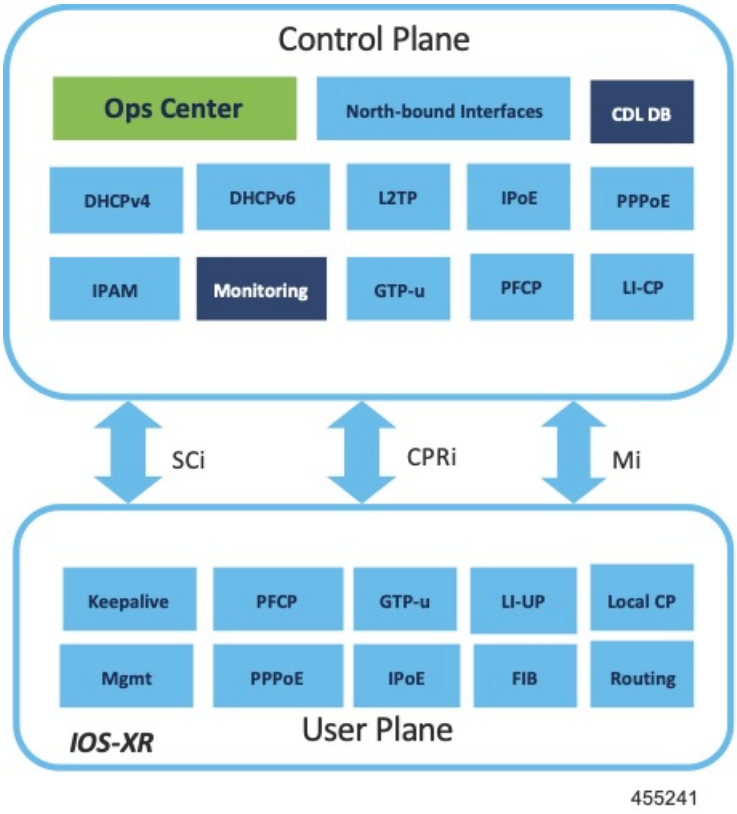
In the cnBNG architecture, the CPs and UPs are clearly and cleanly separated from each other and run in completely distinct and independent environments.

The BNG CP is moved out to a container-based microservice cloud environment.

The UP can be on any of the physical platforms that supports the BNG UP, like Cisco ASR 9000 Series Routers.

The following figure illustrates the overall cnBNG architecture.

*Figure 2: cnBNG Architecture*




### Features and Benefits

The cnBNG supports the following features:

- **Path to convergence**: With shared Subscriber Management infrastructure, common microservices across the policy layer and shared UPs for BNG and Mobile back-haul, cnBNG paves the way for real Fixed Mobile Convergence (FMC).
- **Flexibility of scaling**: cnBNG architecture provides flexibility by decoupling the required scalability dimensions. The CP can be scaled with requirement of number of subscribers to be managed and UPs can be augmented based on the bandwidth requirements. Instead of building the CP for peak usage, the orchestrator can be triggered to deploy the relevant microservices as needed to handle the increased rate of transactions.
- **Distributed UPs**: With reduced operational complexity and minimal integration efforts with centralize CP, UPs can be distributed, closer to end-users to offload traffic to nearest peering points and CDNs. This feature reduces the core transport costs.
- **Cost effective and Leaner User planes**: With the subscriber management functions moved to cloud, you can choose cost-effective UP models for optimized deployment requirements.

The benefits of the cnBNG architecture are:

- Simplified and unified BNG CP
- Platform independent and Network Operation System (NOS) agnostic BNG CP

- Unified Policy interface across both BNG and mobility
- Common infrastructure across wireline and mobility
- Seamless migration from existing deployments
- Leverage the common infrastructure across access technologies
- Standardized model driven interface with the UP
- Data externalization for North-bound interfaces (NBI)
- Highly available and fault tolerant
- Simplified Subscriber Geo redundancy
- Horizontally scalable CP
- Independent CP and UP upgrades
- Feature agility with CI and CD
- Manageability and Operational Simplification

# cnBNG Components

The cnBNG solution comprises of the following components:

## Subscriber Microservices Infrastructure

The Cisco Ultra Cloud Core Subscriber Microservices Infrastructure (SMI) is a layered stack of cloud technologies that enable the rapid deployment, and seamless life-cycle operations for microservices-based applications.

The SMI stack consists of the following:

- SMI Cluster Manager—Creates the Kubernetes (K8s) cluster, creates the software repository, and provides ongoing LCM for the cluster including deployment, upgrades, and expansion.
- Kubernetes Management—Includes the K8s master and etcd functions, which provide LCM for the NF applications deployed in the cluster. This component also provides cluster health monitoring and resources scheduling.
- Common Execution Environment (CEE)—Provides common utilities and OAM functionalities for Cisco cloud native NFs and applications, including licensing and entitlement functions, configuration management, telemetry and alarm visualization, logging management, and troubleshooting utilities. Additionally, it provides consistent interaction and experience for all customer touch points and integration points in relation to these tools and deployed applications.
- Common Data Layer (CDL)—Provides a high performance, low latency, stateful data store, designed specifically for 5G and subscriber applications. This next generation data store offers HA in local or geo-redundant deployments.
- Service Mesh—Provides sophisticated message routing between application containers, enabling managed interconnectivity, additional security, and the ability to deploy new code and new configurations in low risk manner.

- NB Streaming—Provides Northbound Data Streaming service for billing and charging systems.
- NF/Application Worker nodes—The containers that comprise an NF application pod.
- NF/Application Endpoints (EPs)—The NF's/application's interfaces to other entities on the network.
- Application Programming Interfaces (APIs)—SMI provides various APIs for deployment, configuration, and management automation.

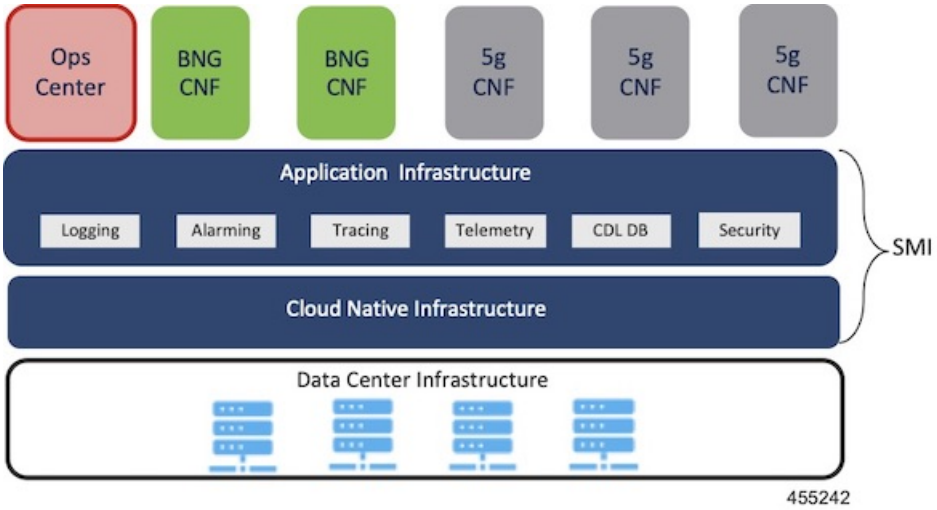
For more information on SMI components, refer to the "Overview" chapter of the *Ultra Cloud Core Subscriber Microservices Infrastructure* documentation—*Deployment Guide*.

For information on the Cisco Ultra Cloud Core, see https://www.cisco.com/c/en/us/products/collateral/wireless/packet-core/datasheet-c78-744630.html.

## cnBNG Control Plane

The Cisco cnBNG CP is built on Cisco® Cloud Native Infrastructure, which is a Kubernetes-based platform that provides a common execution environment for container-based applications. This CP is built on principles of stateless microservices, to scale at-ease, introduce services much faster and more cost-effective.

***Figure 3: cnBNG Control Plane Architecture***




The CP runs as a Virtual Machine (VM) to adapt to existing service provider-deployed virtual infrastructure. It is built ground-up on a clean-slate architecture with a view on ‘Converged Subscriber Services’ and is aligned to 3gpp and BBF standards.

The cnBNG CP effectively manages the subscriber management functions such as:

- Authentication, authorization, and accounting of subscriber sessions
- IP Address assignment
- In-built DHCP Server
- Security
- Policy management
- Quality of Service (QoS)

Service providers can choose from wide choice of available ASR 9000 form factors, based on exact deployment requirements. The CUPS architecture allows to run these UPs in a distributed mode, to the edge of network, for early traffic offloads.

## cnBNG User Plane

The UP delivers the forwarding functionality of the entire cnBNG solution. With the CP handling the subscriber management functionality, the cnBNG architecture enables the UP to be more distributed and interoperable with cnBNG CP with minimal integration efforts. The cnBNG Subscriber Provisioning Agent (SPA), which is the common interface between UP and CP, is bundled with the existing Cisco IOS XR image to transform an integrated physical BNG router to a cnBNG user plane.

For more information about the cnBNG UP, see the *Cloud Native BNG User Plane Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.3.x*.

# License Information

cnBNG supports the following licenses:

| License | Description |
|---|---|
| Application Base | Per cluster |
| Session (Increments) | Network-wide |

These are the software license PIDs for cnBNG:

**Cisco cnBNG Control Plane:**

| Product IDs | Description |
|---|---|
| CN-BNG-BASE-L | Base PID for cnBNG Control Plane (per cluster) |
| CN-BNG-100k-L | Session scale for 100,000 subscribers (network-wide) base licenses |
| CN-BNG-400k-L | Session scale for 400,000 subscribers (network-wide) base licenses |
| CN-BNG-1M-L | Session scale for 1,000,000 subscribers (network-wide) base licenses |
| CN-BNG-2M-L | Session scale for 2,000,000 subscribers (network-wide) base licenses |

**Cisco cnBNG User Planes:**

Refer the ASR9000 data sheet for ordering information:
https://www.cisco.com/c/en/us/products/routers/asr-9000-series-aggregation-services-routers/datasheet-listing.html

# Standard Compliance

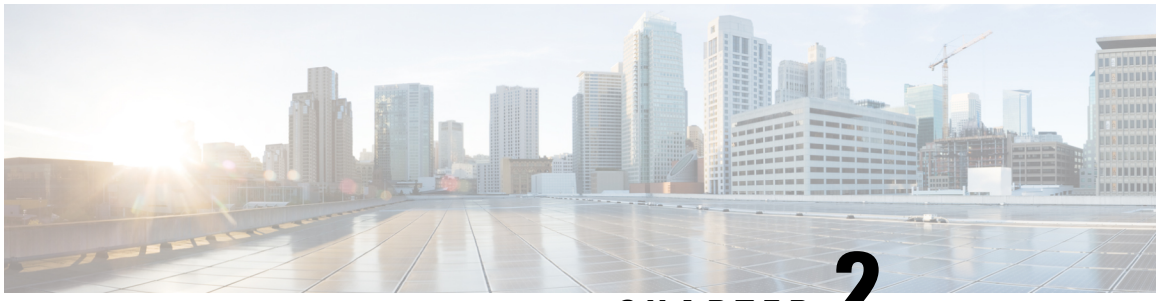
cnBNG solution is aligned with the following standard:

TR-459 Control and User Plane Separation for a disaggregated BNG

# Limitations and Restrictions

The cnBNG has the following limitations and restrictions in this release:

- High availability on CP is not supported.
- Only one subnet is supported per VRF.
- QoS provisioning is supported only through service.



CHAPTER 2

# cnBNG Installation and Configuration



# Feature Summary and Revision History

## Summary Data

***Table 2: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | cnBNG |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Disabled - Configuration Required |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

## Revision History

***Table 3: Revision History***

# Feature Description

This chapter describes cnBNG installation and configuration using the Ultra Cloud Core Subscriber Microservices Infrastructure (SMI) Cluster Manager and the BNG Operations (Ops) Center. The BNG Ops Center is based on the ConfD command line interface (CLI).

To install the SMI Cluster Manager, refer to the "Deploying the SMI Cluster Manager on VMware vCenter" section in the *Ultra Cloud Core Subscriber Microservices Infrastructure - Deployment Guide*.

The SMI Ops Center is the platform to install the cnBNG cluster with the offline or online repository. It is mandatory to install the SMI Ops Center to set up and access the BNG Ops Center.

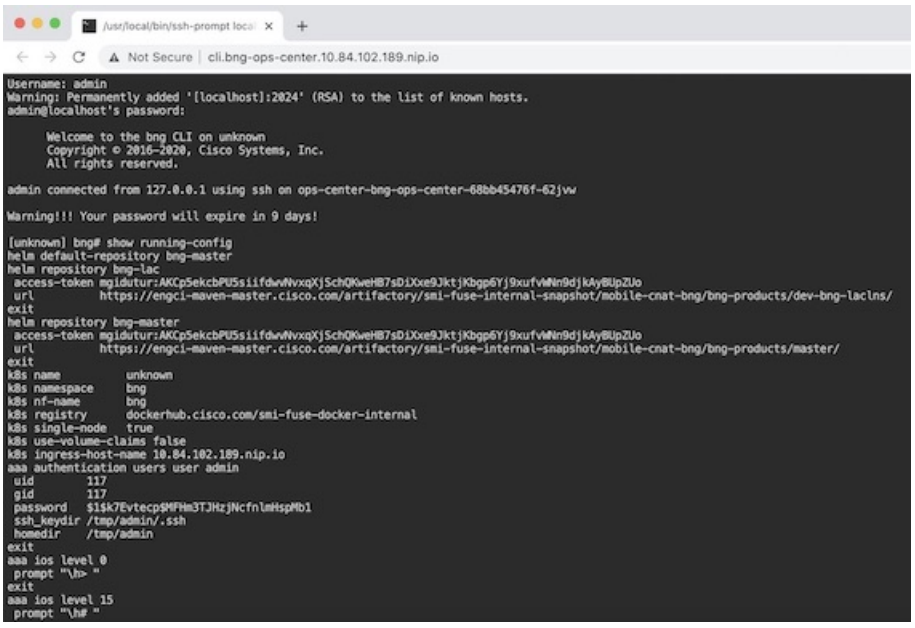
**Note** To access the offline or online repository, contact your Cisco Account Manager or representative to get access to the offline or online repository.

# BNG Ops Center

The BNG Ops Center is a system-level infrastructure that provides the following functionality:

- A user interface to trigger a deployment of microservices with the flexibility of providing variable helm chart parameters to control the scale and properties of Kubernetes objects (deployment, pod, services, and so on) associated with the deployment.
- A user interface to push application-specific configuration to one or more microservices through Kubernetes configuration maps.
- A user interface to issue application-specific execution commands (such as show and clear commands). These commands:
  - Invoke some APIs in application-specific pods
  - Display the information returned on the user interface application

The following figure shows a sample of the web-based CLI presented to the user.




The BNG Ops Center allows you to configure features such as licensing, REST endpoint, and CDL.

# Installing cnBNG and Accessing BNG Ops Center

This section describes how to install cnBNG and access the BNG Ops Center.

The Ultra Cloud Core SMI platform is responsible for setting up and managing the Cloud Native Broadband Network Gateway application.

**Note** The cnBNG installation is tested and qualified on the VMware vCenter 6.7 environment.

## Prerequisites

Before installing cnBNG on the SMI layer in an offline environment:

- Ensure that the SMI Cluster Manager all-in-one (AIO) is installed. This helps orchestrate the K8s Cluster and load the image.
- Ensure that all SMI K8s cluster nodes are in Ready state.
- Run the SMI synchronization operation for the BNG Ops Center and Cloud Native Common Execution Environment (CN-CEE).

  For CEE installation, refer to the *Ultra Cloud Core Common Execution Environment- Configuration and Administration Guide*.
- Ensure that the local repositories, which host the product offline TAR ball version, is installed.

### System Requirements

| Feature | Description |
|---|---|
| Disk Space | 2 x 800 GB SSD (RAID 1) or equivalent input/output operations per second (IOPS) and redundancy. |
| Hardware | • High-performance x86 64-bit chipset<br>• CPU performance Passmark benchmark of 13K rating per chip and 1,365 rating per thread, or better<br>• VMware ESXi-compatible |

| Feature | Description |
|---|---|
| | **Note** The following is recommended:<br>• Cisco UCSM5 series blade servers to achieve the best performance.<br>• All the host servers should be UCSC-C240-M5SX or UCSC-C220-M5SX.<br>• All the UCS systems should have SSD storge type.<br>• UCS C240M5 servers for better performance and to avoid infrastructure issues. |
| Platform | VMware ESXi and VMware vCenter versions 6.5 and 6.7<br>**Note** SMI Cluster Manger support is qualified on the preceding platforms. |
| Memory | • At least DDR3-1600 or better than 1600 MT/s<br>• ECC |
| Deployment Requirement | Hardware oversubscription, network saturation, or CPU oversubscription reduces application performance and productivity. The Cisco Ultra Cloud Core Subscriber Microservices Infrastructure detects and takes action when infrastructure requirements are not met. |

# Installing cnBNG in an Offline Environment

Using the SMI Cluster Manager, download the offline TAR ball of the cnBNG, the host and its charts, and corresponding images in the local registries. The SMI Cluster Manager supports the deployment of the BNG Ops Center and all the applications and services associated with it. This section describes the procedures involved in installing cnBNG in an offline environment using the SMI Cluster Manager.

To install cnBNG, complete the following steps:

1. Download the TAR ball from the URL.

   **software-packages download** *URL*

   **Example**:

   ```
   SMI Cluster Manager# software-packages download
   http://<ipv4address>:<port_number>/packages/bng-2021-02-1.tar
   ```

2. Verify whether the TAR balls are loaded.

   **software-packages list**

   **Example**:

   ```
   BNG Cluster Manager# software-packages list
   [ bng-2021-02-1 ]
   [ sample ]
   ```

3. Configure the necessary SMI Ops Center parameters in the cluster to install cnBNG.

```
config
   cluster cluster_name
      ops-centers app_name instance_name
         repository url
         netconf-ip ipv4_address
         netconf-port port
         ssh-ip ipv4_address
         ssh-port port
         ingress-hostname <ipv4_address>.<customer_specific_domain_name>
         initial-boot-parameters use-volume-claims true/false
         initial-boot-parameters first-boot-password password
         initial-boot-parameters auto-deploy true/false
         initial-boot-parameters single-node true/false
         initial-boot-parameters image-pull-secrets
         exit
   exit
```

**Example**:

```
SMI Cluster Manager# config
Entering configuration mode terminal
SMI Cluster Manager(config)# clusters cnbng-smi-cluster-01
SMI Cluster Manager(config-clusters-cnbng-smi-cluster-01)# ops-centers bng bng
SMI Cluster Manager(config-ops-centers-bng/bng)# repository
https://charts.10.10.105.50.nip.io/bng-2021.02.1
SMI Cluster Manager(config-ops-centers-bng/bng)# ingress-hostname 10.10.105.34.nip.io
SMI Cluster Manager(config-ops-centers-bng/bng)# initial-boot-parameters use-volume-claims
 true
SMI Cluster Manager(config-ops-centers-bng/bng)# initial-boot-parameters
first-boot-password test123
SMI Cluster Manager(config-ops-centers-bng/bng)# initial-boot-parameters auto-deploy
false
SMI Cluster Manager(config-ops-centers-bng/bng)# initial-boot-parameters single-node
false
SMI Cluster Manager(config-ops-centers-bng/bng)# exit
SMI Cluster Manager(config-clusters-cnbng-smi-cluster-01)# exit
SMI Cluster Manager(config)#
```

4. Configure the secrets, if your local registry contains secrets.

```
config
   cluster cluster_name
      secrets docker-registry secret_name
         docker-server server_name
         docker-username username
         docker-password password
         docker-email email
         namespace k8s namespace
         commit
         exit
      exit
```

**Example**:

```
SMI Cluster Manager# config
SMI Cluster Manager(config)# clusters test2
SMI Cluster Manager(config-clusters-test2)# secrets docker-registry sec1
SMI Cluster Manager(config-docker-registry-sec1)# docker-server serv1
SMI Cluster Manager(config-docker-registry-sec1)# docker-username user1
```

```
SMI Cluster Manager(config-docker-registry-sec1)# docker-password Cisco@123
SMI Cluster Manager(config-docker-registry-sec1)# docker-email reg@cisco.com
SMI Cluster Manager(config-docker-registry-sec1)# bng bng
SMI Cluster Manager(config-docker-registry-sec1)# exit
SMI Cluster Manager(config-clusters-test2)# exit
SMI Cluster Manager(config)#
```

5. Run the cluster synchronization.

   **clusters** *cluster_name* **actions sync run**

   **Example**:

   ```
   SMI Cluster Manager# clusters cnbng-smi-cluster-01 actions sync run
   ```

**Notes**:

- **software-packages download** *url*–Specifies the software packages to be downloaded through HTTP/HTTPS.
- **software-packages list**–Specifies the list of available software packages.
- **ops-centers** *app_name instance_name*–Specifies the BNG Ops Center and instance. *app_name* is the application name. *instance_name* is the name of the instance.
- **repository** *url*-Specifies the local registry URL for downloading the charts.
- **netconf-ip** *ipv4_address*–Specifies the BNG Ops Center netconf IPv4 address.
- **netconf-port** *port*–Specifies the BNG Ops Center netconf port number.
- **ssh-ip** *ipv4_address*–Specifies the SSH IPv4 address for the BNG Ops Center.
- **ssh-port** *port*–Specifies the SSH port number for the BNG Ops Center.
- **ingress-hostname** <*ipv4_address*>.<*customer_specific_domain_name*>–Specifies the ingress hostname to be set to the BNG Ops Center. <*customer_specific_domain_name*> specifies the domain name of the customer.
- **initial-boot-parameters**–Specifies the initial boot parameters for deploying the helm charts.
  - **use-volume-claims** *true/false*–Specifies the usage of persistent volumes. Set this option to True to use persistent volumes. The default value is true.
  - **first-boot-password** *password*–Specifies the first boot password for the product's Ops Center.
  - **auto-deploy** *true/false*–Auto deploys all the services of the product. Set this option to false to deploy only the product's Ops Center.
  - **single-node** *true/false*– Specifies the product deployment on a single node. Set this option to false for multi node deployments.
  - **image-pull-secrets**–Specifies the docker registry secret name to be used.
- **secrets docker-registry** *secret_name*–Specifies the secret name for your docker registry.
  - **docker-server** *server_name*–Specifies the docker server name.
  - **docker-username** *username*–Specifies the docker registry user name.
  - **docker-password** *password*–Specifies the docker registry password.

- **docker-email** *email*–Specifies the docker registry email.
- **namespace** *namespace*–Specifies the docker registry namespace.

### Verifying the cnBNG Installation

Verify the status of the cnBNG installation deployment through the cnBNG CLI. To verify, use the following commands:

1. Log in to the cnBNG product CLI.
2. Verify whether the charts are loaded in the specific instance (verify the namespace).

   **show helm charts**

   **Example**:

   ```
   bng# show helm charts
   CHART    INSTANCE  STATUS    VERSION  REVISION  RELEASE   NAMESPACE
   ------------------------------------------------------------------------
   infra-charts - DEPLOYED 0.0.6-rel-2021-01-0073-210208130850-fac5207 1 bng-bng-infra-charts
    bng-bng
   oam-pod - DEPLOYED 0.1.2-rel-2021-01-0144-210122165946-fcb74ed 1 bng-bng-oam-pod bng-bng
   bng-dashboard - DEPLOYED 0.0.1-rel-2021-01-0039-210122165311-0d542be 1
   bng-bng-bng-dashboard bng-bng
   etcd-cluster - DEPLOYED 0.7.0-0-7-0060-210203074532-f118407 1 bng-bng-etcd-cluster bng-bng
   ngn-datastore - DEPLOYED 1.3.0-1-3-0782-210125161812-f50a892 1 bng-bng-ngn-datastore
   bng-bng
   ```

3. Verify the status of the system.

   **show system status**

   **Example**:

   ```
   bng# show system status
   system status deployed true
   system status percent-ready 100.0
   ```

**Notes**:

- **show helm charts**–Displays the helm release details.
- **show system status**–Displays the status of the system.

# Accessing BNG Ops Center

You can connect to the BNG Ops Center through SSH or the web-based CLI console.

1. SSH:

   **ssh admin**@ops_center_pod_ip **-p 2024**

2. Web-based console:
   a. Log in to the Kubernetes master node.
   b. Run the following command:

   ```
   kubectl get ingress <namespace>
   ```

The available ingress connections get listed.

c. Select the appropriate ingress and access the BNG Ops Center.

d. Access the following URL from your web browser:

**cli.**<namespace>**-ops-center.**<ip_address>**.nip.io**

By default, the Day 0 configuration is loaded into the cnBNG.

#### Day 0 Configuration

To view the Day 0 configuration, run the following command.

**show running-config**

The following is a sample Day 0 configuration:

# CP and UP Service Configuration

The CP service requires the basic configuration to process the API calls.

**Note** For information about the User Plane service configuration, refer to the *Cloud Native BNG User Plane Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.3.x*

## Configuring the CP

The CP configuration is provided using the Ops Center infrastructure.

The following is a sample CP configuration:

```
ipam
 source local
 address-pool Default-Pool
  address-quarantine-timer 60
  vrf-name                 default
  ipv4
   split-size
    per-cache 131072
    per-dp    131072
   exit
   address-range 13.0.0.1 13.1.255.255
  exit
  ipv6
   address-ranges
    split-size
     per-cache 65536
     per-dp    65536
    exit
    address-range 1:4::1 1:4::ffff
    address-range 1:5::1 1:5::ffff
    address-range 1:6::1 1:6::ffff
    address-range 1:7::1 1:7::ffff
   exit
   prefix-ranges
    split-size
     per-cache 65536
```

```
    per-dp    65536
   exit
   prefix-range 2003:db0:: length 48
   prefix-range 2003:db1:: length 48
   prefix-range 2003:db2:: length 48
   prefix-range 2003:db3:: length 48
  exit
 exit
exit
address-pool VRF-Pool
 address-quarantine-timer 60
 vrf-name                 it_vrf
 ipv4
  split-size
   per-cache 131072
   per-dp    131072
  exit
  address-range 14.0.0.1 14.1.255.255
 exit
 ipv6
  address-ranges
   split-size
    per-cache 65536
    per-dp    65536
   exit
   address-range 2:4::1 2:4::ffff
   address-range 2:5::1 2:5::ffff
   address-range 2:6::1 2:6::ffff
   address-range 2:7::1 2:7::ffff
  exit
  prefix-ranges
   split-size
    per-cache 65536
    per-dp    65536
   exit
   prefix-range 2004:db0:: length 48
   prefix-range 2004:db1:: length 48
   prefix-range 2004:db2:: length 48
   prefix-range 2004:db3:: length 48
  exit
 exit
exit
address-pool pool-ISP
 address-quarantine-timer 60
 vrf-name                 default
 ipv4
  split-size
   per-cache 131072
   per-dp    131072
  exit
  address-range 11.0.0.1 11.1.255.255
 exit
 ipv6
  address-ranges
   split-size
    per-cache 65536
    per-dp    65536
   exit
   address-range 4:2::1 4:2::ffff
   address-range 4:3::1 4:3::ffff
   address-range 4:4::1 4:4::ffff
   address-range 4:5::1 4:5::ffff
  exit
  prefix-ranges
```

```
    split-size
     per-cache 65536
     per-dp    65536
    exit
    prefix-range 2001:db0:: length 48
    prefix-range 2001:db1:: length 48
    prefix-range 2001:db2:: length 48
    prefix-range 2001:db3:: length 48
   exit
  exit
 exit
 address-pool pool-st
  vrf-name default
  static enable user-plane asr9k-2
  ipv4
   split-size
    per-cache 262144
    per-dp    262144
   exit
   address-range 12.0.0.1 12.3.255.254 default-gateway 12.0.0.1
  exit
  ipv6
   address-ranges
    split-size
     per-cache 8192
     per-dp    8192
    exit
    address-range 2:2::1 2:2::ff00
   exit
   prefix-ranges
    split-size
     per-cache 8192
     per-dp    8192
    exit
    prefix-range 3001:db0:: length 48
   exit
  exit
 exit
 address-pool static-pool
  vrf-name access-vrf-1
  static enable user-plane asr9k-1
  ipv4
   split-size
    no-split
   exit
   address-range 20.20.0.0 20.20.0.255 default-gateway 20.20.0.1
  exit
 exit
exit
cdl node-type session
cdl logging default-log-level error
cdl datastore session
 endpoint replica 2
 endpoint settings slot-timeout-ms 750
 index replica 2
 index map 1
 slot replica 2
 slot map 2
 slot notification limit 300
exit
cdl kafka replica 2
profile dhcp dhcp-server1
 ipv4
  mode server
```

```
  server
   pool-name   pool-ISP
   dns-servers [ 8.8.8.8 ]
   lease hours 6
   lease minutes 40
  exit
 exit
 ipv6
  mode server
  server
   iana-pool-name pool-ISP
   iapd-pool-name pool-ISP
   lease days 0
   lease hours 4
   lease minutes 2
  exit
 exit
exit
profile dhcp dhcp-server3
 ipv4
  mode server
  server
   pool-name   Default-Pool
   dns-servers [ 8.8.8.8 ]
   lease days 1
   lease hours 6
   lease minutes 3
  exit
 exit
 ipv6
  mode server
  server
   iana-pool-name Default-Pool
   iapd-pool-name Default-Pool
   lease days 1
   lease hours 6
   lease minutes 3
  exit
 exit
exit
profile dhcp dhcp-server4
 ipv4
  mode server
  server
   pool-name   VRF-Pool
   dns-servers [ 8.8.8.8 ]
   lease hours 6
   lease minutes 40
  exit
 exit
 ipv6
  mode server
  server
   iana-pool-name VRF-Pool
   iapd-pool-name VRF-Pool
   lease hours 6
  exit
 exit
exit
profile pppoe bng
 ctrl-pkt-priority 7
 max-payload deny
 service-name      [ value]
 ac-name           123@acname
```

```
 ac-cookie         123@accookie
exit
profile aaa aaa-prof1
 authorization
  type subscriber method-order [ local ]
  username value <username>
  password <password>
 exit
 accounting
  method-order [ local ]
 exit
exit
profile server-group local
 radius-group local
exit
profile subscriber subs-default
 dhcp-profile               dhcp-server3
 session-type               ipv4v6
 activate-feature-templates [ svc1 QOS_HSI QOS_IPTV QOS_VOICE ]
 aaa authorize aaa-prof1
exit
profile subscriber subs-prof1
 dhcp-profile               dhcp-server1
 session-type               ipv4v6
 activate-feature-templates [ svc1 ]
 aaa authorize aaa-prof1
exit
profile subscriber subs-prof1-pppoe
 dhcp-profile  dhcp-server1
 pppoe-profile bng
 session-type  ipv4v6
 class ppp_cls_map
  activate-feature-templates [ bng_ft_start ]
  matches
   match-type all
   match protocol [ ppp ]
  exit
 exit
 event session-activate
  class ppp_cls_map
   activate-feature-templates [ bng_ft_activate ]
   matches
    match-type all
    match protocol [ ppp ]
   exit
   aaa authenticate aaa-prof1
  exit
 exit
exit
profile subscriber subs-vrf
 dhcp-profile               dhcp-server4
 session-type               ipv4v6
 activate-feature-templates [ svc3 QOS_VOICE QOS_IPTV QOS_HSI ]
 aaa authorize aaa-prof1
exit
profile subscriber test-ppp-subscriber
 dhcp-profile               dhcp-server3
 pppoe-profile              test-ppp-pppoeprofile
 session-type               ipv4v6
 activate-feature-templates [ svc1 test-ppp-featuretemplate QOS_VOICE QOS_IPTV QOS_HSI ]
 aaa authorize aaa-prof1
exit
profile feature-template ACL-V4
 ipv4
```

```
  ingress-acl iACL_BNG_IPv4_IN
  egress-acl  iACL_BNG_IPv4_OUT
 exit
exit
profile feature-template ACL-V6
 ipv6
  ingress-acl v6-IN
  egress-acl  v6-out
 exit
exit
profile feature-template QOS_HSI
 qos
  in-policy   QOS_HSI_100B_IN
  out-policy  QOS_HSI_100B_OUT
  merge-level 30
 exit
 service-accounting
  enable
  aaa-profile       aaa-prof1
  periodic-interval 1800
 exit
exit
profile feature-template QOS_VOICE
 qos
  in-policy   QOS_VOICE_INGRESS
  out-policy  QOS_VOICE_EGRESS
  merge-level 40
 exit
exit
profile feature-template QOS_IPTV
 qos
  in-policy   QOS_IPTV_INGRESS
  out-policy  QOS_IPTV_EGRESS
  merge-level 50
exit
exit
profile feature-template QOS
 qos
  in-policy   QOS-IN
  out-policy  QOS-OUT
  merge-level 10
 exit
 service-accounting
  enable
  aaa-profile aaa-prof1
 exit
exit
profile feature-template bng_ft_activate
 ipv4
  mtu                  1492
  ingress-acl          in4acl3
  disable-unreachables
  verify-unicast-source reachable-via-rx
 exit
 ipv6
  mtu                  1492
  ingress-acl          match-ipv6-acl
  disable-unreachables
  verify-unicast-source reachable-via-rx
 exit
 session-accounting
  enable
  aaa-profile       aaa-prof1
  periodic-interval 1200
```

```
 exit
 ppp
  ipcp dns 8.8.8.8 1.2.3.4
  ipcp peer-address-pool pool-ISP
  ipcp renegotiation ignore
  ipv6cp renegotiation ignore
 exit
exit
profile feature-template bng_ft_start
 vrf-name default
 session-accounting
  enable
  aaa-profile       aaa-prof1
  periodic-interval 1200
 exit
 ppp
  authentication [ pap ]
  lcp delay seconds 1 milliseconds 0
  lcp renegotiation ignore
 exit
exit
profile feature-template svc1
 vrf-name default
 ipv4
  mtu                  1492
  ingress-acl          iACL_BNG_IPv4_IN_1
  egress-acl           iACL_BNG_IPv4_OUT_1
  disable-unreachables
  verify-unicast-source reachable-via-rx
 exit
 ipv6
  mtu                  1492
  ingress-acl          ipv6-acl-in-1
  egress-acl           ipv6-acl-out-1
  disable-unreachables
  verify-unicast-source reachable-via-rx
 exit
 session-accounting
  enable
  aaa-profile       aaa-prof1
  periodic-interval 1800
 exit
exit
profile feature-template svc2
 ppp
  ipcp peer-address-pool poolv4
  ipcp renegotiation ignore
  lcp renegotiation ignore
 exit
exit
profile feature-template svc3
 vrf-name it_vrf
 ipv4
  mtu                  1492
  ingress-acl          iACL_BNG_IPv4_IN_1
  egress-acl           iACL_BNG_IPv4_OUT_1
  disable-unreachables
  verify-unicast-source reachable-via-rx
 exit
 ipv6
  mtu                  1492
  ingress-acl          ipv6-acl-in-1
  egress-acl           ipv6-acl-out-1
  disable-unreachables
```

```
  verify-unicast-source reachable-via-rx
 exit
 session-accounting
  enable
  aaa-profile       aaa-prof1
  periodic-interval 1800
 exit
exit
profile feature-template svc4
 vrf-name default
 session-accounting
  enable
  aaa-profile       aaa-prof1
  periodic-interval 1800
 exit
exit
profile feature-template test-ppp-featuretemplate
 vrf-name default
 ipv4
  mtu 1400
 exit
 ppp
  ipcp peer-address-pool Default-Pool
  ipcp renegotiation ignore
  ipv6cp renegotiation ignore
  lcp renegotiation ignore
 exit
exit
profile feature-template uRPF
 ipv4
  verify-unicast-source reachable-via-rx
 exit
 ipv6
  verify-unicast-source reachable-via-rx
 exit
exit
profile radius
 algorithm round-robin
 deadtime  3
 detect-dead-server response-timeout 60
 max-retry 1
 timeout   5
 server 172.16.254.55 1812
  type   auth
  secret <secret_value>
 exit
 server 172.16.254.55 1813
  type   acct
  secret <secret_value>
 exit
 server 172.16.254.56 1812
  type   auth
  secret <secret_value>
 exit
 server 172.16.254.56 1813
  type   acct
  secret <secret_value>
 exit
 attribute
  nas-identifier < any identifier>
  nas-ip         172.16.254.86
  nas-port-id < add_unique_id>
 exit
 server-group local
```

```
  server auth 172.16.254.55 1812
  exit
  server auth 172.16.254.56 1812
  exit
  server acct 172.16.254.55 1813
  exit
  server acct 172.16.254.56 1813
  exit
 exit
exit
profile coa
 client 172.16.254.55
  server-key < key >
 exit
 client 172.16.254.56
  server-key < key >
 exit
exit
user-plane <add UP name like asr9k-11>
 peer-address ipv4 172.16.247.72
 subscriber-profile subs-default
exit
endpoint sm
exit
endpoint nodemgr
exit
endpoint n4-protocol
exit
endpoint dhcp
exit
endpoint radius
 replicas 1
 vip-ip 172.16.254.86
 interface coa-nas
  sla response 140000
  vip-ip 172.16.254.86 vip-port 2000
 exit
exit
endpoint udp-proxy
 replicas 1
 nodes    2
 vip-ip 172.16.254.86 vip-port 3799
 interface n4
  sla response 150000
 exit
 interface gtpu
  sla response 150000
 exit
exit
endpoint charging
exit
logging transaction duplicate enable
logging name bng-dhcp0.bngfsol.collision level application info
logging name bng-dhcp0.bngfsol.collision level transaction info
logging name infra.application.core level application warn
logging name infra.config.core level application error
logging name infra.config.core level transaction error
k8 bng
 etcd-endpoint      etcd:2379
 datastore-endpoint datastore-ep-session:8882
 tracing
  enable
  enable-trace-percent 30
  append-messages      true
```

```
  endpoint             jaeger-collector:9411
 exit
exit
k8 label protocol-layer key smi.cisco.com/vm-type value protocol
exit
k8 label service-layer key smi.cisco.com/vm-type value service
exit
k8 label cdl-layer key smi.cisco.com/vm-type value session
exit
k8 label oam-layer key smi.cisco.com/vm-type value oam
exit
system mode running
exit
```

## Configuring the UP

The following is a sample UP configuration:

```
user-plane asr9k-11
peer-address ipv4 10.105.247.124
subscriber-profile subs-default
port-id Bundle-Ether2.10
  subscriber-profile subs-vrf
exit
port-id Bundle-Ether2.20
  subscriber-profile subs-vrf
port-id Bundle-Ether2.10
exit
port-id Bundle-Ether2.30
  subscriber-profile subs-vrf
port-id Bundle-Ether2.10
exit
port-id Bundle-Ether2.40
  subscriber-profile subs-vrf
port-id Bundle-Ether2.10
exit
exit
```

## Loading Day1 Configuration

To load the Day 1 configuration for cnBNG, run the following command:

```
ssh admin@ops_center_pod_ip -p 2024  < Day1config.cli
```

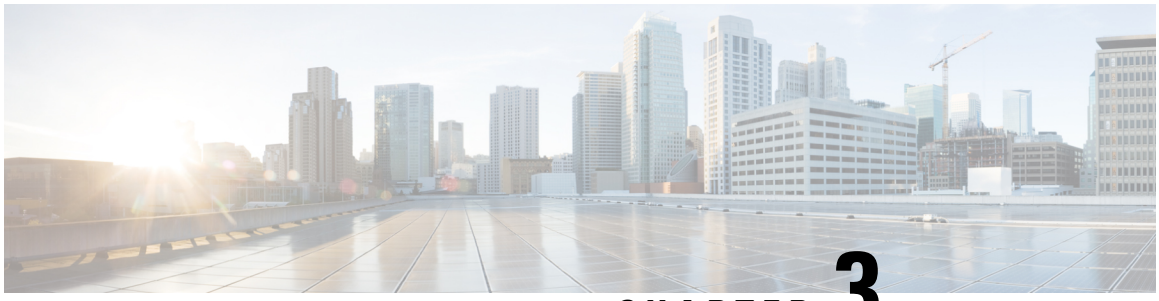
**Note** The **day1config.cli file** contains the necessary parameters required for the Day 1 configuration.

Alternatively, you can copy the configuration and paste it in the BNG Ops Center CLI to load the Day 1 configuration.

```
config
  <Paste the Day 1 configuration here>
  commit
  exit
```

### Day1config.cli

The **day1config.cli file** file contains the Day 1 configuration for cnBNG. For a sample day1 configuration, see Configuring the CP, on page 16.



CHAPTER 3

# Pods and Services Reference



# Feature Summary and Revision History

## Summary Data

***Table 4: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | cnBNG |
| Applicable Platform(s) | SMI |
| Feature Default Setting | |
| Related Documentation | Not Applicable |

## Revision History

***Table 5: Revision History***

| Revision Details | Release |
|---|---|
| First introduced. | 2021.01.0 |

# Feature Description

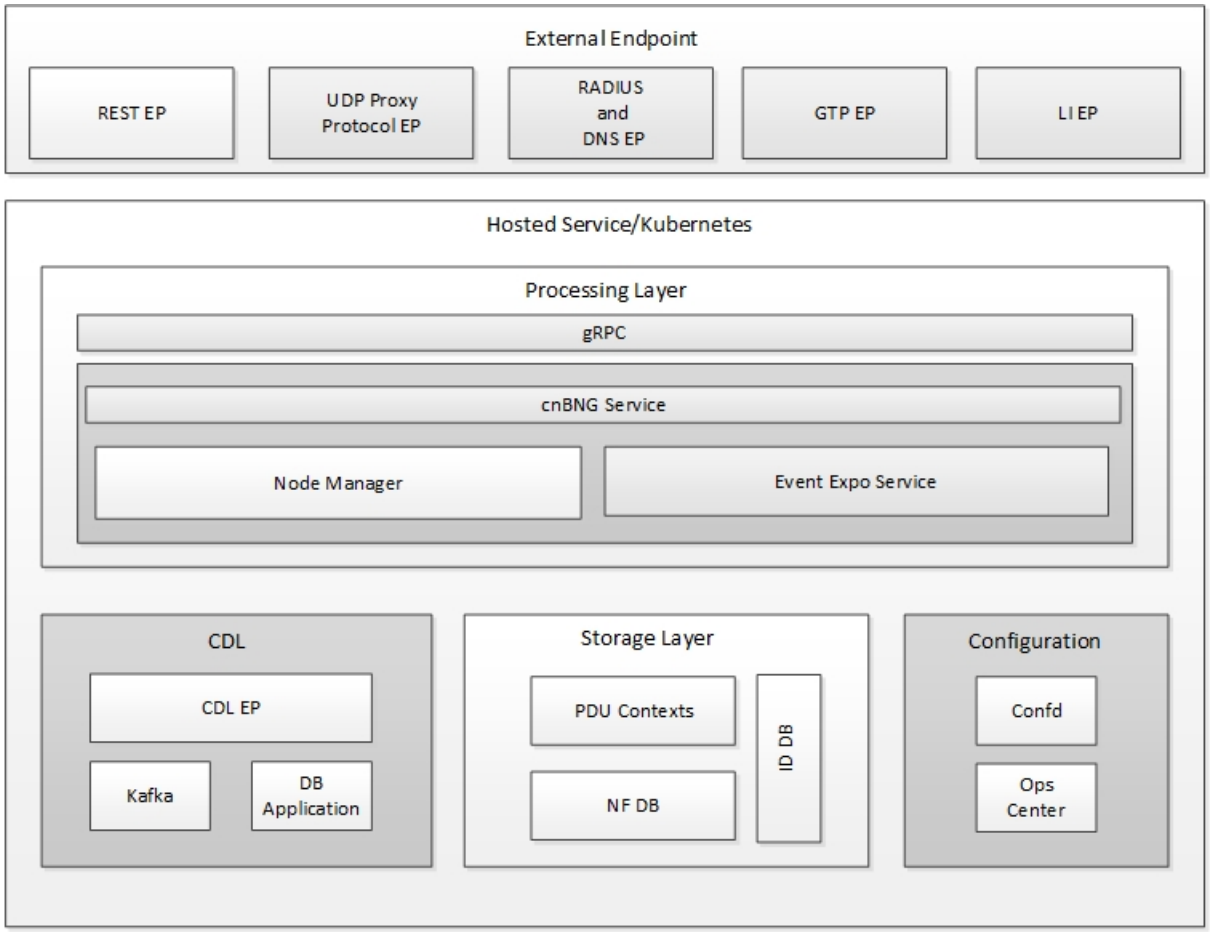
The SMI Ops Center is the platform to deploy cnBNG cluster with the offline or online repository. It is mandatory to deploy the SMI Ops Center to install the BNG Ops Center.

The cnBNG is built on the Kubernetes cluster strategy, which implies that it has adopted the native concepts of containerization, high availability, scalability, modularity, and ease of deployment. To achieve the benefits offered by Kubernetes, cnBNG uses the construct that includes the components such as pods and services.

Depending on the deployment environment, the cnBNG deploys the pods on the virtual machines that you have configured. Pods operate through the services that are responsible for the intra-pod communications. If the machine hosting the pods fail or experiences network disruption, the pods are terminated or deleted. However, this situation is transient and BNG spins new pods to replace the invalid pods.

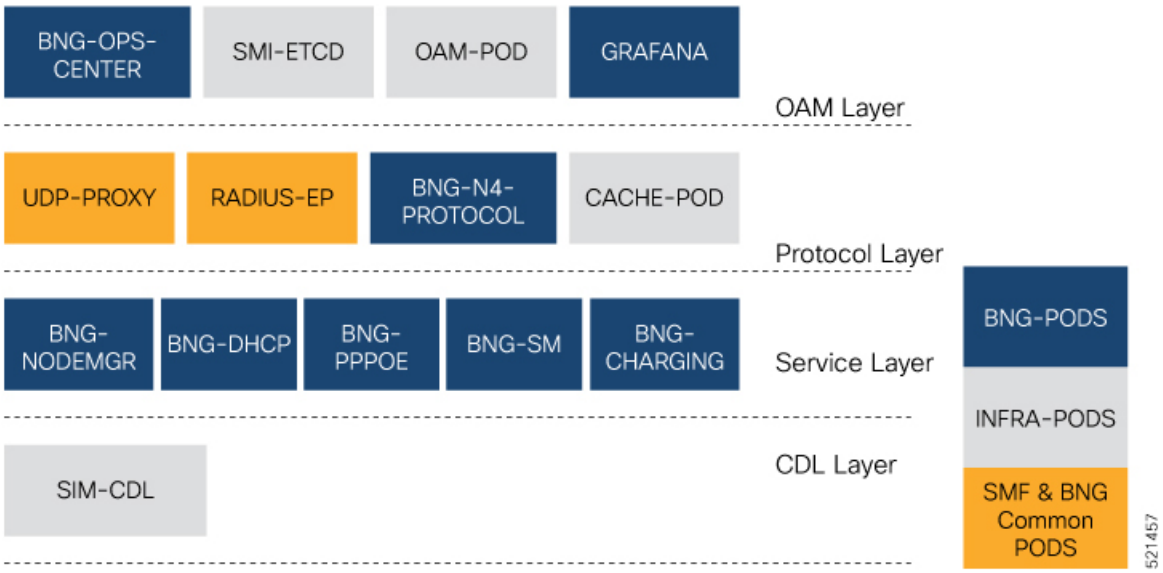
The following workflow provides a high-level visibility into the host machines, and the associated pods and services. It also represents how the pods interact with each other. The representation might defer based on your deployment infrastructure.

***Figure 4: Communication Workflow of Pods***




The following figure shows the cnBNG cluster pod layout.

*Figure 5: cnBNG Cluster Pod Layout*




Kubernetes deployment includes the **kubectl** command-line tool to manage the Kubernetes resources in the cluster. You can manage the pods, nodes, and services.

For generic information on the Kubernetes concepts, see the Kubernetes documentation.

The following sections provide more information on the Kubernetes components in cnBNG.

# Pods

A pod is a process that runs on your Kubernetes cluster. It encapsulates a granular unit known as a container. A pod contains one or multiple containers.

Kubernetes deploys one or multiple pods on a single node which can be a physical or virtual machine. Each pod has a discrete identity with an internal IP address and port space. However, the containers within a pod can share the storage and network resources.

The following tables list the cnBNG and Common Execution Environment (CEE) pod names and the hosts on which they are deployed depending on the labels that you assign. For information on how to assign the labels, see Associating Pods to the Nodes, on page 33.

*Table 6: cnBNG Pods*

| Pod Name | Description | Host Name |
|---|---|---|
| api-bng-bng-ops-center | Functions as the *confD* API pod for the BNG Ops Center. | OAM |
| bng-charging-n0 | Enables subscriber session and service charging for BNG subscribers. | Service |
| bng-dhcp-n0 | Operates as the DHCP server and handles all DHCP related control messages. | Service |

| Pod Name | Description | Host Name |
|---|---|---|
| bng-n4-protocol-n0 | Operates as encoder and decoder of application protocols (PFCP, GTP, RADIUS, and so on) whose underlying transport protocol is UDP. | Protocol |
| bng-nodemgr-n0 | Performs node level interactions Service such as N4 link establishment, management (heart-beat), and so on. | Service |
| bng-pppoe-n0 | Runs the combined Control Plane (CP) for PPPoE and PPP. | Service |
| bng-sm-n0 | Enables subscriber binding with the BNG CP protocol and services. | Service |
| cache-pod-0 | Operates as the pod to cache any sort of system information that will be used by other pods as applicable. | Protocol |
| cdl-ep-session-c1-d0 | Provides an interface to the CDL. | Session |
| cdl-index-session-c1-m1-0 | Preserves the mapping of keys to the session pods. | Session |
| cdl-slot-session-c1-m1-0 | Operates as the CDL Session pod Session to store the session data. | Session |
| documentation | Contains the documentation. | OAM |
| etcd-bng-bng-etcd-cluster-0 | Hosts the etcd for the BNG application to store information such as pod instances, leader information, NF-UUID, endpoints, and so on. | OAM |
| grafana-dashboard-app-infra | Contains the default dashboard of app-infra metrics in Grafana. | OAM |
| grafana-dashboard-bng | Contains the default dashboard of the cnBNG-service metrics in Grafana. | OAM |
| grafana-dashboard-cdl | Contains the default dashboard of CDL metrics in Grafana. | OAM |
| kafka | Hosts theKafka details for the CDL replication. | Protocol |
| oam-pod | Operates as the pod to facilitate Ops Center actions like show commands, configuration commands, monitor protocol monitor subscriber, and so on. | OAM |
| ops-center-bng-bng-ops-center | Acts as the BNG Ops Center. | OAM |
| prometheus-rules-cdl | Contains the default alerting rules and recording rules for Prometheus CDL. | OAM |
| radius-ep-n0-0 | Operates as RADIUS endpoint of cnBNG. | |
| smart-agent-bng-bng-ops-center | Operates as the utility pod for the BNG Ops Center. | OAM |

| Pod Name | Description | Host Name |
|---|---|---|
| bng-udp-proxy-0 | Operates as proxy for all UDP messages. Owns UDP client and server functionalities. | Protocol |
| swift-bng-bng-ops-center | Operates as the utility pod for the BNG Ops Center. | OAM |
| zookeeper | Assists Kafka for topology management. | OAM |

***Table 7: CEE Pods***

| Pod Name | Description | Host Name |
|---|---|---|
| alert-logger | Stores the history of active and resolved alerts. | OAM |
| alertmanager | Duplicates alerts and sends out resolution of alerts when they are resolved in Prometheus. | OAM |
| api-cee-global-ops-center | Functions as the confD API pod for the CEE Ops Center. | OAM |
| bulk-stats | Assists to retrieve bulkstats saved by Prometheus containers. | OAM |
| cee-global-product-documentation | Contains the product documentation (API, CLI, and so on). | OAM |
| core-retriever | Assists in retrieving the core dumps. | All the nodes except ETCD nodes. |
| documentation | Contains the documentation (metrics and usage). | OAM |
| grafana-dashboard-metrics | Assists in collating Grafana metrics on the dashboard. | OAM |
| grafana | Contains the Grafana metrics for CEE. | OAM |
| kube-state-metrics | Assists in generating metrics about the state of Kubernetes objects: node status, node capacity (CPU and memory), and so on. | OAM |
| logs-retriever | Assists in retrieving Kernel, Kubelet, and Container level logs through output to JournalD driver. | All the nodes except ETCD nodes. |
| node-exporter | Exports the node metrics. | All the nodes. |
| ops-center-cee-global-ops-center | Provides NETCONF and CLI interface to the application. | OAM |
| path-provisioner | Provisions the local storage volume. | All the nodes except ETCD nodes. |

| Pod Name | Description | Host Name |
|---|---|---|
| pgpool | *Pgpool* is a middleware that works between *PostgreSQL* servers and a *PostgreSQL* database. | OAM |
| postgres | Storage of alerts and Grafana dashboards. | OAM |
| prometheus-hi-res | Stores all metrics and generates alerts by alerting rules. | OAM |
| prometheus-rules | Contains the default alerting rules and recording rules for Prometheus. | OAM |
| prometheus-scrapeconfigs-synch | Synchronizes the Prometheus scrape configuration. | OAM |
| pv-manager | Provisions the local storage volume. | OAM |
| pv-provisioner | Provisions the local storage volume. | OAM |
| show-tac-manager | Assists in creating and deleting debug package. | OAM |
| smart-agent-cee-global-ops-center | Operates as the utility pod for the CEE Ops Center. | OAM |
| snmp-trapper | Sends the SNMP traps based on triggered alerts. | OAM |
| swift-cee-global-ops-center | Operates as the utility pod for the CEE Ops Center. | OAM |
| thanos-query-hi-res | Implements the Thanos query for Prometheus HA. | OAM |
| fluentbit | Assists in log forwarding to the external logs collector. | All the nodes except ETCD nodes. |

# Services

The cnBNG configuration is composed of several microservices that run on a set of discrete pods. Microservices are deployed during the cnBNG deployment. The cnBNG uses these services to enable communication between the pods. When interacting with another pod, the service identifies the IP address of the pod to initiate the transaction and acts as an endpoint for the pod.

The following table describes the BNG services and the pod on which they run.

***Table 8: BNG Services and Pods***

| Service Name | Pod Name | Description |
|---|---|---|
| bng-nodemgr | bng-nodemgr-n0 | Responsible for node level interactions Service such as N4 link establishment, management (heart-beat), and so on. |
| bng-charging | bng-charging-n0 | Responsible for subscriber session and service charging for BNG subscribers. |

| Service Name | Pod Name | Description |
|---|---|---|
| bng-dhcp | bng-dhcp-n0 | Functions as the DHCP server and handles all DHCP related control messages. |
| bng-pppoe | bng-pppoe-n0 | Functions as the combined Control Plane (CP) for PPPoE and PPP. |
| bng-sm | bng-sm-n0 | Responsible for the subscriber binding with the BNG CP protocol and services. |

## Open Ports and Services

cnBNG uses different ports for communication purposes. The following table describes the default open ports and the associated services in an SMI based cnBNG system.

### Application Infrastructure (App-infra)

| Port | Service |
|---|---|
| 8850 | Golang net/HTTP server TCP Golang net/HTTP server |
| 8879 | Golang net/HTTP server TCP Golang net/HTTP server |
| 8850 | DefaultPProfPort |
| 8879 | DefaultAdminEndPointPort |

### UDP

| Port | Service | CP to UP Interfaces |
|---|---|---|
| 2152 | GTPU | CPRi |
| 8805 | PFCP | SCi |

# Associating Pods to the Nodes

This section describes how to associate a pod to the node based on their labels.

After you have configured a cluster, you can associate pods to the nodes through labels. This association enables the pods to get deployed on the appropriate node based on the key-value pair.

Labels are required for the pods to identify the nodes where they must get deployed and to run the services. For example, when you configure the protocol-layer label with the required key-value pair, the pods are deployed on the nodes that match the key-value pair.

To associate pods to the nodes through the labels, use the following configuration:

1. To associate pods to the nodes through the labels, use the following configuration:

```
config
k8 label protocol-layer key key_value vm-type value protocol
exit
k8 label service-layer key key_value vm-type value service
exit
k8 label cdl-layer key key_value vm-type value cdl
exit
k8 label oam-layer key key_value vm-type value oam
exit
```

**NOTES:**

- If you opt not to configure the labels, then BNG assumes the labels with the default key-value pair.
  - **k8 label protocol-layer key** *key_value* **vm-type** *value* **protocol**: Configures the key value pair for protocol layer.
  - **k8 label service-layer key** *key_value* **vm-type** *value* **service**: Configures the key value pair for the service layer.
  - **k8 label cdl-layer key** *key_value* **vm-type** *value* **cdl**: Configures the key value pair for CDL.
  - **k8 label oam-layer key** *key_value* **vm-type** *value* **oam**: Configures the key value pair for OAM layer.

# Viewing the Pod Details and Status

If the service requires additional pods, BNG creates and deploys the pods. You can view the list of pods that are participating in your deployment through the BNG Ops Center.

You can run the **kubectl** command from the master node to manage the Kubernetes resources.

1. To view the comprehensive pod details, use the following command.

   ```
   kubectl get pods -n bng_namespace pod_name -o yaml
   ```

   The pod details are available in YAML format. The output of this command results in the following information:

   - The IP address of the host where the pod is deployed.
   - The service and application that is running on the pod.
   - The ID and name of the container within the pod.
   - The IP address of the pod.
   - The current state and phase in which the pod is.
   - The start time from which pod is in the current state.

2. Use the following command to view the summary of the pod details.

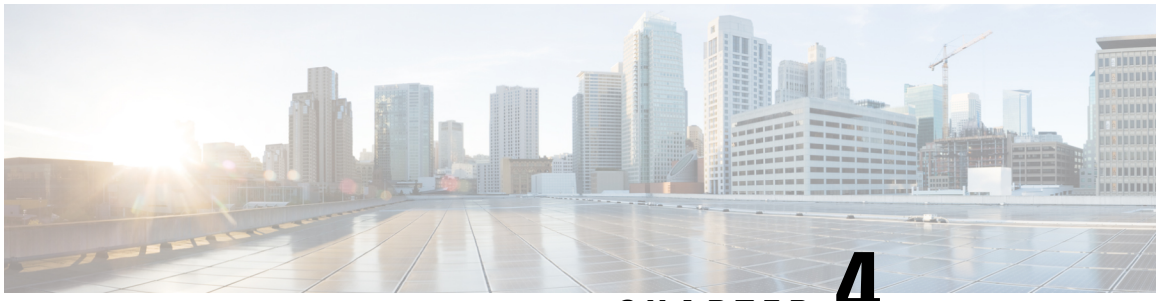
   ```
   kubectl get pods -n bng_namespace -o wide
   ```

# States

Understanding the pod's state lets you determine the current health and prevent the potential risks. The following table describes the pod's states.

***Table 9: Pod States***

| State | Description |
|---|---|
| Running | The pod is healthy and deployed on a node.<br>It contains one or more containers. |
| Pending | The application is in the process of creating the container images for the pod. |
| Succeeded | Indicates that all the containers in the pod are successfully terminated. These pods cannot be restarted. |
| Failed | One ore more containers in the pod have failed the termination process. The failure occurred as the container either exited with non zero status or the system terminated the container. |
| Unknown | The state of the pod could not be determined. Typically, this could be observed because the node where the pod resides was not reachable. |



CHAPTER 4

# Cisco Common Data Layer



# Feature Summary and Revision History

## Summary Data

***Table 10: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | cnBNG |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Disabled - Configuration Required |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

## Revision History

***Table 11: Revision History***

| Revision Details | Release |
|---|---|
| First introduced. | 2021.01.0 |

# Feature Description

The Cisco Common Data Layer (CDL) is a high-performance next generation Key-value (KV) data store layer for all the Cloud Native applications. These applications use the CDL as a state management with High Availability (HA) and Geo HA functions. The CDL provides:

- Different Network Functions (NFs) - such as AMF, cnBNG Control Plane, SMF, and PCF - microservices.
- Multi-master support to achieve low latency read and write.
- Pure in-memory storage.
- Session related timers to notify NF on timer expiry.

Deploying CDL provides the following benefits:

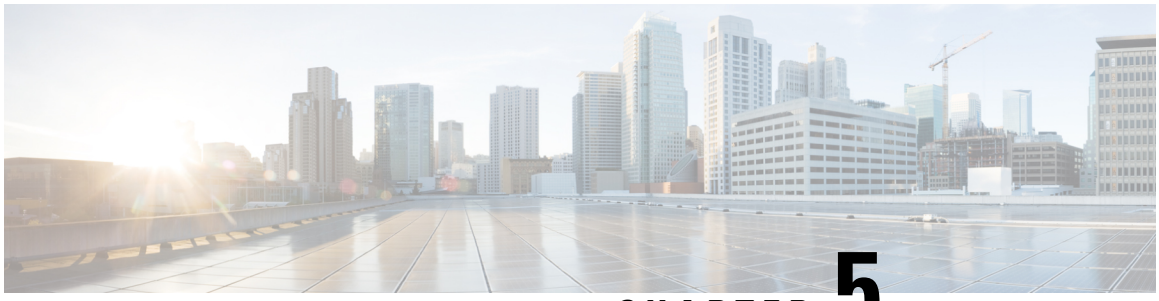
- Service-Based Architecture (SBA) with auto discovery and global accessibility.
- High performance, in-memory caching and in-memory data store.
- Container-based solution from the ground up.
- CDL can deploy and scale with simple API calls.
- Geo Redundant Replication among multiple cnBNG clusters.

For detailed information about CDL, refer to the *UCC SMI Common Data Layer Configuration and Administration Guide* at https://www.cisco.com/c/en/us/support/wireless/ultra-cloud-core-subscriber-microservices-infrastructure/products-installation-and-configuration-guides-list.html.

# Limitations

The CDL feature has the following limitation on cnBNG.

Geo-redundancy is not supported.



CHAPTER 5

# Authentication, Authorization, and Accounting Functions



# Feature Summary and Revision History

## Summary Data

***Table 12: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | cnBNG |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Disabled - Configuration Required |
| Related Changes in this Release | Not Applicable |
| Related Documentation | *Cloud Native BNG Control Plane Command Reference Guide* |

## Revision History

***Table 13: Revision History***

| Revision Details | Release |
|---|---|
| First introduced. | 2021.01.0 |

# Feature Description

**Note:** All references to BNG in this chapter refer to the Cloud-Native Broadband Network Gateway (cnBNG).

This chapter provides information about configuring authentication, authorization, and accounting (AAA) functions on the BNG. BNG interacts with the RADIUS server to perform AAA functions. A group of RADIUS servers form a server group that is assigned specific AAA tasks. A method list defined on a server or server group lists methods by which authorization is performed. Some of the RADIUS features include creating specific AAA attribute formats, load balancing of RADIUS servers, throttling of RADIUS records, Change of Authorization (CoA), Session Accounting, and Service Accounting for QoS.

## AAA Overview

AAA acts as a framework for effective network management and security. It helps in managing network resources, enforcing policies, auditing network usage, and providing bill-related information. BNG connects to an external RADIUS server that provides the AAA functions.

The RADIUS server performs the three independent security functions (authentication, authorization, and accounting) to secure networks against unauthorized access. The RADIUS server runs the Remote Authentication Dial-In User Service (RADIUS) protocol. (For details about RADIUS protocol, refer to RFC 2865). The RADIUS server manages the AAA process by interacting with BNG, and databases and directories containing user information.

The RADIUS protocol runs on a distributed client-server system. The RADIUS client runs on BNG (Cisco ASR 9000 Series Router) that sends authentication requests to a central RADIUS server. The RADIUS server contains all user authentication and network service access information.

The AAA processes, the role of RADIUS server during these processes, and some BNG restrictions, are explained in these sections:

### Authentication

The authentication process identifies a subscriber on the network, before granting access to the network and network services. The process of authentication works on a unique set of criteria that each subscriber has for gaining access to the network. Typically, the RADIUS server performs authentication by matching the credentials (user name and password) the subscriber enters with those present in the database for that subscriber. If the credentials match, the subscriber is granted access to the network. Otherwise, the authentication process fails, and network access is denied.

### Authorization

After the authentication process, the subscriber is authorized for performing certain activity. Authorization is the process that determines what type of activities, resources, or services a subscriber is permitted to use. For example, after logging into the network, the subscriber may try to access a database, or a restricted website. The authorization process determines whether the subscriber has the authority to access these network resources.

AAA authorization works by assembling a set of attributes based on the authentication credentials provided by the subscriber. The RADIUS server compares these attributes, for a given username, with information contained in a database. The result is returned to BNG to determine the actual capabilities and restrictions that are to be applied for that subscriber.

### Accounting

The accounting keeps track of resources used by the subscriber during network access. Accounting is used for billing, trend analysis, tracking resource utilization, and capacity planning activities. During the accounting process, a log is maintained for network usage statistics. The information monitored include, but are not limited to - subscriber identities, applied configurations on the subscriber, the start and stop times of network connections, and the number of packets and bytes transferred to, and from, the network.

BNG reports subscriber activity to the RADIUS server in the form of accounting records. Each accounting record comprises of an accounting attribute value. This value is analyzed and used by the RADIUS server for network management, client billing, auditing, etc.

The accounting records of the subscriber sessions may timeout if the BNG does not receive acknowledgments from the RADIUS server. This timeout can be due to RADIUS server being unreachable or due to network connectivity issues leading to slow performance of the RADIUS server. It is therefore recommended that a RADIUS server **deadtime** be configured on the BNG, to avoid loss of sessions. Once this value is configured, and if a particular session is not receiving an accounting response even after retries, then that particular RADIUS server is considered to be non-working and further requests are not sent to that server.

### Restrictions

- On session disconnect, transmission of the Accounting-Stop request to RADIUS may be delayed for a few seconds while the system waits for the "final" session statistics to be collected from the hardware. The Event-Timestamp attribute in that Accounting-Stop request should, however, reflect the time the client disconnects, and not the transmission time.

# Using RADIUS Server Group

A RADIUS server group is a named group of one or more RADIUS servers. Each server group is used for a particular service. For example, in an AAA network configuration having two RADIUS server groups, the first server group can be assigned the authentication and authorization task, while the second group can be assigned the accounting task.

Server groups can include multiple host entries for the same server. Each entry, however, must have a unique identifier. This unique identifier is created by combining an IP address and a UDP port number. Different ports of the server, therefore, can be separately defined as individual RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on the same server. Further, if two different host entries on the same RADIUS server are configured for the same service (like the authentication process), then the second host entry acts as a fail-over backup for the first one. That is, if the first host entry fails to provide authentication services, BNG tries with the second host entry. (The RADIUS host entries are tried in the order in which they are created.)

For assigning specific actions to the server group, see Configuring RADIUS Server Group, on page 58.

# Specifying Method Order

Method order for AAA defines the methods using which authorization is performed, and the sequence in which these methods are executed. Before any defined authentication method is performed, the method order must be applied to the configuration mechanism responsible for validating user-access credentials.

On BNG, you have to specify the method order and the server group that will be used for AAA services. For specifying method order, see Configuring Method Order for AAA, on page 49.

# Defining AAA Attributes

The AAA attribute is an element of RADIUS packet. A RADIUS packet transfers data between a RADIUS server and a RADIUS client. The AAA attribute parameter, and its value - form a Attribute Value Pair (AVP). The AVP carries data for both requests and responses for the AAA transaction.

The AAA attributes either can be predefined as in Internet Engineering Task Force (IETF) attributes or vendor defined as in vendor-specific attributes (VSAs). For more information about the list of BNG supported attributes, see RADIUS Attributes, on page 137.

The RADIUS server provides configuration updates to BNG in the form of attributes in RADIUS messages. The configuration updates can be applied on a subscriber during session setup through two typical methods—per-user attributes, which applies configuration on a subscriber as part of the subscriber's authentication Access Accept, or through explicit domain, port, or service authorization Access Accepts. This is all controlled by the Policy Rule Engine's configuration on the subscriber.

When BNG sends an authentication or an authorization request to an external RADIUS server as an Access Request, the server sends back configuration updates to BNG as part of the Access Accept. In addition to RADIUS configuring a subscriber during setup, the server can send a change of authorization (CoA) message autonomously to the BNG during the subscriber's active session life cycle, even when the BNG did not send a request. These RADIUS CoA updates act as dynamic updates, referencing configured elements in the BNG and instructing the BNG to update a particular control policy or service policy.

BNG supports the concept of a "service", which is a group of configured features acting together to represent that service. Services can be represented as either features configured on dynamic-templates through CLI, or as features configured as RADIUS attributes inside Radius Servers. Services are activated either directly from CLI or RADIUS through configured "activate" actions on the Policy Rule Engine, or through CoA "activate-service" requests. Services can also be deactivated directly (removing all the involved features within the named service) through configured "deactivate" action on the Policy Rule Engine or through CoA "deactivate-service" requests.

The attribute values received from RADIUS interact with the subscriber session in this way:

- BNG merges the values received in the RADIUS update with the existing values that were provisioned statically by means of CLI commands, or from prior RADIUS updates.
- In all cases, values received in a RADIUS update take precedence over any corresponding CLI provisioned values or prior RADIUS updates. Even if you reconfigured the CLI provisioned values, the system does not override session attributes or features that were received in a RADIUS update.
- Changes made to CLI provision values on the dynamic template take effect immediately on all sessions using that template, assuming the template features have not already been overridden by RADIUS. Same applies to service updates made through CoA "service-update" requests.

### AAA Attribute Format

It is possible to define a customized format for some attributes. For the configuration syntax for creating a new format, see Configuring AAA Attributes, on page 48.

Once the format is defined, the FORMAT-NAME can be applied to various AAA attributes such as username, nas-port-ID, calling-station-ID, and called-station-ID. The configurable AAA attributes that use the format capability are explained in the section Creating Attributes of Specific Format, on page 43.

To create a customized nas-port attribute and apply a predefined format to nas-port-ID attribute , see Configuring RADIUS Attribute Format, on page 54.

Specific functions can be defined for an attribute format for specific purposes. For example, if the input username is "text@abc.com", and only the portion after "@" is required as the username, a function can be defined to retain only the portion after "@" as the username. Then, "text" is dropped from the input, and the new username is "abc.com". To apply username truncation function to a named-attribute format, see Configuring AAA Attributes, on page 48.

## Creating Attributes of Specific Format

BNG supports the use of configurable AAA attributes. The configurable AAA attributes have specific user-defined formats. The following sections list some of the configurable AAA attributes used by BNG.

### Username

BNG has the ability to construct AAA username and other format-supported attributes for subscribers using MAC address, circuit-ID, remote-ID, and DHCP Option-60 (and a larger set of values available in CLI). The DHCP option-60 is one of the newer options that is communicated by the DHCP client to the DHCP server in its requests; it carries Vendor Class Identifier (VCI) of the DHCP client's hardware.

The MAC address attribute is specified in the CLI format in either of these forms:

- mac-address: for example, 0000.4096.3e4a
- mac-address-ietf: for example, 00-00-40-96-3E-4A
- mac-address-raw: for example, 000040963e4a
- mac-address-custom1: for example, 01.23.45.67.89.AB

  (This particular MAC address format is available only from Release 6.2.1 and later).

### NAS-Port-ID

The NAS-Port-ID is constructed by combining BNG port information and access-node information. The BNG port information consists of a string in this form:

```
"eth phy_slot/phy_subslot/phy_port:XPI.XCI"
```

For 802.1Q tunneling (QinQ), XPI is the outer VLAN tag and XCI is the inner VLAN tag.

If the interface is QinQ, the default format of nas-port-ID includes both the VLAN tags; if the interface is single tag, it includes a single VLAN tag.

In the case of a single VLAN, only the outer VLAN is configured, using this syntax:

```
<slot>/<subslot>/<port>/<outer_vlan>
```

In the case of QinQ, the VLAN is configured using this syntax:

```
<slot>/<subslot>/<port>/<inner_vlan>.<outer_vlan>
```

In the case of a bundle-interface, the phy_slot and the phy_subslot are set to zero (0); whereas the phy_port number is the bundle number. For example, 0/0/10/30 is the NAS-Port-ID for a Bundle-Ether10.41 with an outer VLAN value 30.

The nas-port-ID command is extended to use the 'nas-port-type' option so that the customized format (configured with the command shown above) can be used on a specific interface type (nas-port-type).

If 'type' option is not specified, then the nas-port-ID for all interface types is constructed according to the format name specified in the command.

### Calling-Station-ID and Called-Station-ID

BNG supports the use of configurable calling-station-ID and called-station-ID. The calling-station-ID is a RADIUS attribute that uses Automatic Number Identification (ANI), or similar technology. It allows the network access server (NAS) to send to the Access-Request packet, the phone number from which the call came from. The called-station-ID is a RADIUS attribute that uses Dialed Number Identification (DNIS), or similar technology. It allows the NAS to send to the Access-Request packet, the phone number that the user called from.

# Making RADIUS Server Settings

In order to make BNG interact with the RADIUS server, certain server specific settings must be made on the BNG router.

For more making RADIUS server settings, see Configuring RADIUS Server, on page 58.

### Restriction

The service profile push or asynchronously pushing a profile to the system is not supported. To download a profile from Radius, the profile must be requested initially as part of the subscriber request. Only service-update is supported and can be used to change a service that was previously downloaded.

# Balancing Transaction Load on the RADIUS Server

The RADIUS load-balancing feature is a mechanism to share the load of RADIUS access and accounting transactions, across a set of RADIUS servers. Each AAA request processing is considered to be a transaction. BNG distributes batches of transactions to servers within a server group.

When the first transaction for a new is received, BNG determines the server with the lowest number of outstanding transactions in its queue. This server is assigned that batch of transactions. BNG keeps repeating this determination process to ensure that the server with the least-outstanding transactions always gets a new batch. This method is known as the least-outstanding method of load balancing.

For configuring the load balancing on the RADIUS server, see Configuring RADIUS Server Selection Logic, on page 58.

# RADIUS Change of Authorization Overview

The RADIUS Change of Authorization (CoA) function allows the RADIUS server to change the authorization settings for a subscriber who is already authorized. CoA is an extension to the RADIUS standard that allows sending asynchronous messages from RADIUS servers to a RADIUS client, like BNG.

**Note** A CoA server can be a different from the RADIUS server.

To identify the subscriber whose configuration needs to be changed, a RADIUS CoA server supports and uses a variety of keys (RADIUS attributes) such as Accounting-Session-ID, Username, IP-Address, and ipv4:vrf-id.

The RADIUS CoA supports:

- account-update — BNG parses and applies the attributes received as part of the CoA profile. Only subscriber-specific attributes are supported and applied on the user profile.
- activate-service — BNG starts a predefined service on a subscriber. The service settings can either be defined locally by a dynamic template, or downloaded from the RADIUS server.
- deactivate-service — BNG stops a previously started service on the subscriber, which is equivalent to deactivating a dynamic-template.

For a list of supported Vendor-Specific Attributes for account operations, see Vendor-Specific Attributes for Account Operations, on page 142.

**Note** In order for BNG to enable interim accounting, it is mandatory for the CoA request to have both accounting method list from the dynamic-template and Acct-Interim-Interval attribute from the user profile. This behavior is applicable for accounting enabled through dynamic-template. Whereas, from Cisco IOS XR Software Release 5.3.0 and later, the CoA request needs to have only the Acct-Interim-Interval attribute in the user profile.

### Service Activate from CoA

BNG supports activating services through CoA requests. The CoA **service-activate** command is used for activating services. The CoA request for the service activate should contain these attributes:

- "subscriber:command=activate-service" Cisco VSA
- "subscriber:service-name=<service name>" Cisco VSA
- Other attributes that are part of the service profile

The "<subscriber:sa=<service-name>" can also be used to activate services from CoA and through RADIUS.

Duplicate service activate requests can be sent to BNG from the CoA server. BNG does not take any action on services that are already activated. BNG sends a CoA ACK message to the CoA server under these scenarios:

- When a duplicate request with identical parameters comes from the CoA for a service that is already active.
- When a duplicate request with identical parameters comes from the CoA to apply a parameterized service.

BNG sends a CoA NACK message to the CoA server with an error code as an invalid attribute under these scenarios:

- When a request comes from the CoA to deactivate a non-parameterized service that is not applied to the session.
- When a request comes from the CoA to deactivate a parameterized service that is not applied to the session.
- When a duplicate request to apply a parameterized service is made with non-identical parameters from the CoA.
- When a request with non-identical parameters comes from CoA to deactivate a parameterized service.

### Service Update from CoA

The service update feature allows an existing service-profile to be updated with a new RADIUS attribute list representing the updated service. This impacts any subscriber who is already activated with the service and new subscriber who activate the service in the future. The new CoA **service-update** command is used for activating this feature. The CoA request for the service update should have these attributes:

- "subscriber:command=service-update" Cisco VSA
- "subscriber:service-name=<service name>" Cisco VSA
- Other attributes that are part of the service profile

A service update CoA should have a minimum of these attributes:

- vsa cisco generic 1 string "subscriber:command=service-update"
- vsa cisco generic 1 string "subscriber:service-name=<service name>"

### Web Logon with RADIUS Based CoA

To support Web Logon, a set of Policy Rule Events need to be configured in an ordered manner. These events are as follows:

- session-start:
  - On the start of a session, a subscriber is setup to get internet connectivity. The service is activated to redirect HTTP traffic to a Web portal for web-based logon.
  - Start the timer with duration for the maximum waiting period for authentication.
- account-logon—The Web portal collects the user credentials such as username and password and triggers a CoA account-logon command. When this event is triggered, subscriber username and password are authenticated by the RADIUS server. Once the authentication is successful, the HTTP redirect service is deactivated, granting user access to already connected internet setup. Also, the timer established in session-start must be stopped. However, if the authentication fails during account-logon, BNG sends a NAK CoA request, allowing for further authentication attempts to take place.
- timer expiry—When the timer expires, the subscriber session is disconnected based on the configuration.

# User Authentication and Authorization in the Local Network

The user authentication and authorization in the local network feature in BNG provides the option to perform subscriber authorization locally (in a subscriber's network), instead of both remote authentication and authorization that occurs in RADIUS servers. With the User Authentication and Authorization in the Local Network feature, you can run the RADIUS server locally in your network, manage, and configure the RADIUS server locally in your network to the profile that is required for the environment. In the case of a remote RADIUS server, the RADIUS server is maintained by an external regulatory body (not within the subscriber's network) and subscriber will not be able to manage or configure the server.

# Service Accounting

Accounting records for each service enabled on a subscriber can be sent to the configured RADIUS server. These records can include service-start, service-stop, and service-interim records containing the current state

of the service and any associated counters. This feature is the Service Accounting feature. Service accounting records are consolidated accounting records that represent the collection of features that make up a service as part of a subscriber session.

For more information on service accounting for QoS, refer to Authentication, Authorization, and Accounting Functions, on page 39. For more information on commands to configure service accounting, refer to the Configuring Service Accounting, on page 134.

## Standard Compliance

The AAA features are aligned with the following standards:

- RFC 2865 - Remote Authentication Dial In User Service (RADIUS)
- RFC 2866 - RADIUS Accounting
- RFC 5176 - Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)

# Configuring AAA Functions

This section describes how to configure the following Authentication, Authorization, and Accounting (AAA) functions on the Control Plane (CP).

The configuration of the AAA functions involves the following procedures:

- Configuring AAA Attributes
- Configuring the CoA-NAS Interface
- Configuring Method Order for AAA
- Configuring RADIUS Accounting Options
- Configuring RADIUS Accounting Server Group
- Configuring RADIUS Attributes
- Configuring RADIUS-Dead Time
- Configuring RADIUS-Detect Dead Server
- Configuring RADIUS Pod
- Configuring RADIUS Maximum Retry
- Configuring RADIUS NAS-IP
- Configuring RADIUS Server
- Configuring RADIUS Server Group
- Configuring RADIUS Server Selection Logic
- Configuring RADIUS Timeout

# Configuring AAA Attributes

Use the following commands to configure a function for the AAA attribute format.

**NOTES:**

- **profile attribute-format** *attribute_format_name*: Specifies the AAA attributes and enters the Attribute Format Configuration mode.
- **authorization**: Enters the Authorization sub-mode.
- **format-order** *attribute_format* **| identifier { addr | circuit-id-tag | client-mac-address | client-mac-address-custom1 | client-mac-address-custom2 | client-mac-address-ietf | client-mac-address-raw | dhcp-client-id | dhcp-client-id-spl | dhcp-user-class | dhcp-vendor-class | dhcpv4-client-id-spl | dhcpv4-vendor-class | dhcpv6-client-id-ent-ident | dhcpv6-interface-id | dhcpv6-vendor-class-string | inner-vlan-id | outer-vlan-id | physical-adapter | physical-chassis | physical-port | physical-slot | physical-subslot | port-type | pppoe-session-id | remote-id-tag | service-name | username } | value** *value* **}**: Specifies the AAA attribute format order as follows:
  - **addr**: Specifies the IPv4 address of the subscriber.
  - **circuit-id-tag**: Specifies the circuit identifier tag.
  - **client-mac-address**: Specifies the client MAC address in AABB.CCDD.EEFF format.
  - **client-mac-address-custom1**: Specifies the first custom client MAC address in AABB.CCDD.EEFF format.
  - **client-mac-address-custom2**: Specifies the second custom client MAC address in AABB.CCDD.EEFF format.
  - **client-mac-address-ietf**: Specifies the client MAC address in Internet Engineering Task Force (IETF) format. That is, AA-BB-CC-DD-EE-FF format.
  - **client-mac-address-raw**: Specifies the client MAC address in raw (AABBCCDDEEFF) format.
  - **dhcp-client-id**: Specifies the DHCP client identifier.
  - **dhcp-client-id-spl**: Specifies the DHCP client identifier special string.
  - **dhcp-user-class**: Specifies the DHCP user class.
  - **dhcp-vendor-class**: Specifies the DHCP vendor class.
  - **dhcpv4-client-id-spl**: Specifies the DHCPv4 client identifier special string.
  - **dhcpv4-vendor-class**: Specifies the DHCPv4 vendor class.
  - **dhcpv6-client-id-ent-ident**: Specifies the DHCPv6 client and enterprise identifiers.
  - **dhcpv6-interface-id**: Specifies the DHCPv6 interface identifier.
  - **dhcpv6-vendor-class-string**: Specifies the DHCPv6 vendor class string.
  - **inner-vlan-id**: Specifies the inner VLAN identifier.
  - **outer-vlan-id**: Specifies the outer VLAN identifier.
  - **physical-adapter**: Specifies the physical adapter.

- **physical-chassis**: Specifies the physical chassis.
- **physical-port**: Specifies the physical port.
- **physical-slot**: Specifies the physical slot.
- **physical-subslot**: Specifies the physical subslot.
- **port-type**: Specifies the interface or port type.
- **pppoe-session-id**: Specifies the PPPoE physical identifier.
- **remote-id-tag**: Specifies the remote identifier tag.
- **service-name**: Specifies the service name.
- **username**: Specifies the username.

# Configuring the CoA-NAS Interface

Use the following configuration to define Change of Authorization (CoA) NAS interface in the RADIUS endpoint.

```
config
   endpoint radius
      interface coa-nas
         vip-ip ipv4_address vip-port port_number
         end
```

**NOTES:**

- **endpoint radius**: Enters the RADIUS endpoint configuration mode.
- **interface coa-nas**: This keyword defines a new interface "coa-nas", and allows to enter the CoA NAS interface configuration mode.
- **vip-ip** *ipv4_address* **vip-port** *port_number*: Configures the IP address of the host. *ipv4_address* must be in standard IPv4 dotted decimal notation.

  You can configure a list of VIP-IPs to listen to the inbound CoA or DM requests.

  **vip-port** *port_number*: Specify the port number of the UDP proxy. By default, the port number is 3799. This default value is used only when the VIP-IP is specified.

**Important** This configuration allows only port to be specified per IP.

The BNG (udp-pxy) listens to the inbound CoA or DM request messages on these ports and ACK or NAK messages sent with the respective source ip and port.

# Configuring Method Order for AAA

Use the following commands to assign the method order for the server group to use for subscriber authentication, authorization, and accounting.

### Authentication

```
config
  profile aaa aaa_name
    authentication
      method-order custom_server_group
      commit
```

**NOTES:**

- **profile aaa** *aaa_name*: Specifies the AAA profile name and enters the AAA Configuration mode.
- **authentication**: Enters the Authentication sub-mode.
- **method-order** *custom_server_group*: Specifies the method-order to be applied by default for subscriber authentication.

  *custom_server_group* specifies the name of the server group where the method-order is applied.

### Authorization

```
config
  profile aaa aaa_name
    authorization
      password password
      type subscriber method-order custom_server_group
     username { format attribute_format | identifier { addr | circuit-id-tag
 | client-mac-address | client-mac-address-custom1 |
client-mac-address-custom2 | client-mac-address-ietf |
client-mac-address-raw | dhcp-client-id | dhcp-client-id-spl |
dhcp-user-class | dhcp-vendor-class | dhcpv4-client-id-spl |
dhcpv4-vendor-class | dhcpv6-client-id-ent-ident | dhcpv6-interface-id |
 dhcpv6-vendor-class-string | inner-vlan-id | outer-vlan-id |
physical-adapter | physical-chassis | physical-port | physical-slot |
physical-subslot | port-type | pppoe-session-id | remote-id-tag |
service-name | username } | value value }
      commit
```

**NOTES:**

- **profile aaa** *aaa_name*: Specifies the AAA profile name and enters the AAA Configuration mode.
- **authorization**: Enters the Authorization sub-mode.
- **password** *password* : Specifies the password for subscriber authentication.
- **type subscriber method-order** *custom_server_group*: Specifies the method-order to be applied by default for subscriber authorization.

  *custom_server_group* specifies the name of the server group where the method-order is applied.
- **username { format** *attribute_format* **| identifier { addr | circuit-id-tag | client-mac-address | client-mac-address-custom1 | client-mac-address-custom2 | client-mac-address-ietf | client-mac-address-raw | dhcp-client-id | dhcp-client-id-spl | dhcp-user-class | dhcp-vendor-class | dhcpv4-client-id-spl | dhcpv4-vendor-class | dhcpv6-client-id-ent-ident | dhcpv6-interface-id | dhcpv6-vendor-class-string | inner-vlan-id | outer-vlan-id | physical-adapter | physical-chassis |**

**physical-port | physical-slot | physical-subslot | port-type | pppoe-session-id | remote-id-tag | service-name | username } | value** *value* **}**: Specifies the username format, identifier, or value.

- **format** *attribute_format*: Specifies the username attribute format.
- **identifier { addr | circuit-id-tag | client-mac-address | client-mac-address-custom1 | client-mac-address-custom2 | client-mac-address-ietf | client-mac-address-raw | dhcp-client-id | dhcp-client-id-spl | dhcp-user-class | dhcp-vendor-class | dhcpv4-client-id-spl | dhcpv4-vendor-class | dhcpv6-client-id-ent-ident | dhcpv6-interface-id | dhcpv6-vendor-class-string | inner-vlan-id | outer-vlan-id | physical-adapter | physical-chassis | physical-port | physical-slot | physical-subslot | port-type | pppoe-session-id | remote-id-tag | service-name | username }**: Specifies the username identifiers as follows:
  - **addr**: Specifies the IPv4 address of the subscriber.
  - **circuit-id-tag**: Specifies the circuit identifier tag.
  - **client-mac-address**: Specifies the client MAC address in AABB.CCDD.EEFF format.
  - **client-mac-address-custom1**: Specifies the first custom client MAC address in AABB.CCDD.EEFF format.
  - **client-mac-address-custom2**: Specifies the second custom client MAC address in AABB.CCDD.EEFF format.
  - **client-mac-address-ietf**: Specifies the client MAC address in Internet Engineering Task Force (IETF) format. That is, AA-BB-CC-DD-EE-FF format.
  - **client-mac-address-raw**: Specifies the client MAC address in raw (AABBCCDDEEFF) format.
  - **dhcp-client-id**: Specifies the DHCP client identifier.
  - **dhcp-client-id-spl**: Specifies the DHCP client identifier special string.
  - **dhcp-user-class**: Specifies the DHCP user class.
  - **dhcp-vendor-class**: Specifies the DHCP vendor class.
  - **dhcpv4-client-id-spl**: Specifies the DHCPv4 client identifier special string.
  - **dhcpv4-vendor-class**: Specifies the DHCPv4 vendor class.
  - **dhcpv6-client-id-ent-ident**: Specifies the DHCPv6 client and enterprise identifiers.
  - **dhcpv6-interface-id**: Specifies the DHCPv6 interface identifier.
  - **dhcpv6-vendor-class-string**: Specifies the DHCPv6 vendor class string.
  - **inner-vlan-id**: Specifies the inner VLAN identifier.
  - **outer-vlan-id**: Specifies the outer VLAN identifier.
  - **physical-adapter**: Specifies the physical adapter.
  - **physical-chassis**: Specifies the physical chassis.
  - **physical-port**: Specifies the physical port.
  - **physical-slot**: Specifies the physical slot.
  - **physical-subslot**: Specifies the physical subslot.

- **port-type**: Specifies the interface or port type.
- **pppoe-session-id**: Specifies the PPPoE physical identifier.
- **remote-id-tag**: Specifies the remote identifier tag.
- **service-name**: Specifies the service name.
- **username**: Specifies the username.

### Accounting

```
config
  profile aaa aaa_name
    accounting
      method-order custom_server_group
      commit
```

**NOTES:**

- **profile aaa** *aaa_name*: Specifies the AAA profile name and enters the AAA Configuration mode.
- **accounting**: Enters the Accounting sub-mode.
- **method-order** *custom_server_group*: Specifies the method-order to be applied by default for subscriber accounting.

  *custom_server_group* specifies the name of the server group where the method-order is applied.

# Configuring RADIUS Accounting Options

This section describes how to configure the RADIUS accounting options.

**NOTES**:

- **profile radius accounting**: Enters the RADIUS accounting configuration mode.
- **algorithm { first-server | round-robin }**: Defines the algorithm for selecting the RADIUS server.
  - **first-server**: Sets the selection logic as highest priority first. This is the default behavior.
  - **round-robin**: Sets the selection logic as round-robin order of servers.
- **attribute { nas-identifier** *value* **| nas-ip** *ipv4_address* **}** : Configures the RADIUS identification parameters.
  - **nas-identifier** *value*: Specifies the attribute name by which the system will be identified in Accounting-Request messages. *value* must be an alphanumeric string.
  - **nas-ip** *ipv4_address*: Specifies the NAS IPv4 address. *ipv4_address* must be an IPv4 address in dotted decimal notation.
- **deadtime** *value*: Sets the time to elapse between RADIUS server marked unreachable and when we can re-attempt to connect.

  *value* must be an integer from 0 through 65535. Default: 10 minutes.

- **detect-dead-server response-timeout** *value*: Sets the timeout value that marks a server as "dead" when a packet is not received for the specified number of seconds.

  *value* must be an integer from 1 through 65535. Default: 10 seconds.

- **max-retry** *value*: Sets the maximum number of times that the system will attempt retry with the RADIUS server.

  *value* must be an integer from 0 through 65535. Default: 2

- **timeout** *value*: Sets the time to wait for response from the RADIUS server before retransmitting.

  *value* must be an integer from 1 through 65535. Default: 2 seconds.

- **commit**: Commits the configuration.
- All the keyword options under the RADIUS accounting configuration mode are also available within the RADIUS configuration mode.

# Configuring RADIUS Accounting Server Group

This section describes how to configure the RADIUS server group.

```
configure
  profile radius
    server-group group_name
    commit
```

**NOTES**:

- **profile radius**: Enters the RADIUS configuration mode.
- **server group** *group_name*: Specifies the name of server group for use in RADIUS accounting. *group_name* must be an alphanumeric string.
- **commit**: Commits the configuration.

# Configuring RADIUS Attributes

This section describes how to configure the RADIUS attributes for authentication and acocunting.

```
config
  profile radius
    attribute { nas-identifier value | nas-ip ipv4_address }
    commit
```

**NOTES**:

- **profile radius**: Enters the RADIUS configuration mode.
- **attribute { nas-identifier** *value* **| nas-ip** *ipv4_address* **}** : Configures the RADIUS identification parameters.
  - **nas-identifier** *value*: Specifies the attribute name by which the system will be identified in Accounting-Request messages. *value* must be an alphanumeric string.

- **nas-ip** *ipv4_address*: Specifies the NAS IPv4 address. *ipv4_address* must be an IPv4 address in dotted decimal notation.

- **commit**: Commits the configuration.

### Sample Configuration

The following is a sample configuration.

```
config
   profile radius
      attribute
      nas-identifier Ciscobng
      exit
   exit
```

# Configuring RADIUS Attribute Format

# Configuring RADIUS Dead Time

This section describes how to configure the RADIUS dead time.

```
config
   profile radius
      deadtime value
      commit
```

**NOTES**:

- **profile radius**: Enters the RADIUS configuration mode.
- **deadtime** *value*: Sets the time to elapse between RADIUS server marked unreachable and when an reattempt to connect can be made.

  *value* must be an integer from 0 through 65535. Default: 10 minutes.

- **commit**: Commits the configuration.

### Sample Configuration

The following is a sample configuration.

```
config
   profile radius
     deadtime 15
     exit
```

## Configuring RADIUS Detect Dead Server

This section describes how to configure the RADIUS detect dead server.

```
config
   profile radius
      detect-dead-server response-timeout value
      commit
```

**NOTES**:

- **profile radius**: Enters the RADIUS configuration mode.
- **detect-dead-server response-timeout** *value*: Sets the timeout value that marks a server as "dead" when a packet is not received for the specified number of seconds.

  *value* must be an integer from 1 through 65535. Default: 10 seconds.
- **commit**: Commits the configuration.

### Sample Configuration

The following is a sample configuration.

```
config
   profile radius
      detect-dead-server response-timeout 100
      exit
```

# Configuring RADIUS NAS-IP

This section describes how to configure the RADIUS NAS-IP.

### Global RADIUS NAS-IP Configuration

**Important** This configuration is obsolete in 2020.02.x, 2021.1.0 and later releases.

Use the following configuration to configure the NAS-IP address.

```
config
   endpoint radius-dns
      interface radius-client
         vip-ip ipv4_address
         commit
```

**NOTES:**

- **endpoint radius-dns**: Enters the endpoint radius-ep configuration mode.
- **interface radius-client**: Enters the radius-client interface-type configuration mode.
- **vip-ip** *ipv4_address*: Sets the NAS-IP value, which is also used as the source-IP in UDP requests towards the RADIUS server.
- **commit**: Commits the configuration.

### Configuration Example:

```
config
   endpoint radius-dns
      interface radius-client
      vip-ip 209.165.200.228
      exit
   exit
exit
```

### Multiple RADIUS NAS-IP Configuration

Use the following configuration to configure multiple RADIUS NAS-IP addresses at various levels.

```
config
  profile radius
     attribute nas-ip-address ipv4_address
     accounting attribute nas-ip-address ipv4_address
     server-group group_name attribute nas-ip-address ipv4_address
     server-group group_name accounting attribute nas-ip-address ipv4_address

     commit
```

**NOTES:**

- **profile radius**: Enters the RADIUS accounting configuration mode.
- **attribute nas-ip-address** *ipv4_address*: Sets the global NAS-IP address value.
- **accounting attribute nas-ip-address** *ipv4_address*: Sets the global accounting NAS-IP address value.
- **server-group** *group_name* **attribute nas-ip-address** *ipv4_address*: Sets the per server-group common NAS-IP address value.
- **server-group** *group_name* **accounting attribute nas-ip-address** *ipv4_address*: Sets the per server-group accounting NAS-IP address value.
- **commit**: Commits the configuration.

**Configuration Example:**

```
config
 profile radius
  attribute
   nas-ip-address 209.165.200.233
  exit
  accounting
   attribute
    nas-ip-address 209.165.200.235
   exit
  exit
  server-group grp1
   attribute
    nas-ip-address 209.165.200.236
   exit
   accounting
   attribute
    nas-ip-address 209.165.200.237
   exit
  exit
 server-group grp2
   attribute
    nas-ip-address 209.165.200.241
   exit
   accounting
   attribute
    nas-ip-address 209.165.200.239
   exit
  exit
 exit
exit
```

# Configuring RADIUS Pod

This section describes how to configure the RADIUS pod.

```
config
   endpoint radius
      replicas number_of_replicas
      commit
```

**NOTES**:

- **endpoint radius**: Enters the RADIUS endpoint configuration mode.
- **replicas** *number_of_replicas*: Sets the number of replicas required.
- **commit**: Commits the configuration.

### Sample Configuration

The following is a sample configuration.

```
config
   endpoint radius
      replicas 3
      exit
```

# Configuring RADIUS Retries

This section describes how to configure the maximum RADIUS retries.

```
config
   profile radius
      max-retry value
      commit
```

**NOTES**:

- **profile radius**: Enters the RADIUS configuration mode.
- **max-retry** *value*: Sets the maximum number of times that the system will attempt retry with the RADIUS server.

  *value* must be an integer from 0 through 65535. Default: 2
- **commit**: Commits the configuration.

### Sample Configuration

The following is a sample configuration.

```
config
   profile radius
      max-retry 2
      exit
```

# Configuring RADIUS Server

This section describes how to configure the RADIUS server settings.

```
config
  profile radius
    server ipv4_address port_number
      secret secret_key
      priority priority_value
      type { acct | auth }
      commit
```

**NOTES**:

- **profile radius**: Enters the RADIUS configuration mode.
- **server** *ipv4_address port_number*: Specifies the IPv4 address and port of the RADIUS server.
- **secret** *secret_key*: Specifies the secret key.
- **priority** *priority_value*: Specifies the server priority.
- **type { acct | auth }**: Specifies the type of the RADIUS server. It can be one of the following:
  - acct: RADIUS server used for the accounting requests
  - auth: RADIUS server used for the authentication requests
- **commit**: Commits the configuration.

# Configuring RADIUS Server Group

Use the following commands to configure the RADIUS server group.

```
config
  profile server-group server_group_name
    radius-group radius_server_group_name
    commit
```

**NOTES:**

- **profile server-group** *server_group_name*: Specifies the profile server group name to enter the Profile Server Group Configuration mode.
- **radius-group** *radius_server_group_name*: Specifies the RADIUS group server name.

# Configuring RADIUS Server Selection Logic

This section describes how to configure the RADIUS server selection logic.

```
config
  profile radius
    algorithm { first-server | round-robin }
    commit
```

**NOTES**:

- **profile radius**: Enters the RADIUS configuration mode.
- **algorithm { first-server | round-robin }**: Defines the algorithm for selecting the RADIUS server.
  - **first-server**: Sets the selection logic as highest priority first. This is the default behavior.
  - **round-robin**: Sets the selection logic as round-robin order of servers.
- **commit**: Commits the configuration.

### Sample Configuration

The following is a sample configuration.

```
config
   profile radius
      algorithm round-robin
      exit
```

# Configuring RADIUS Timeout

This section describes how to configure the RADIUS timeout.

```
config
   profile radius
      timeout value
      commit
```

**NOTES**:

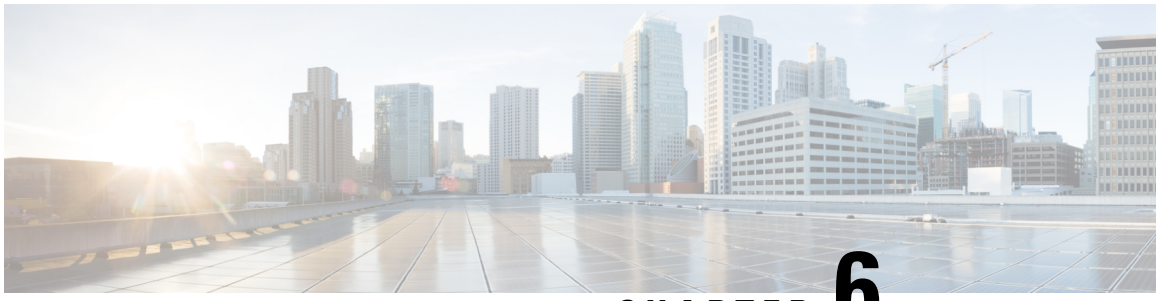
- **profile radius**: Enters the RADIUS configuration mode.
- **timeout** *value_in_seconds*: Sets the time to wait for response from the RADIUS server before retransmitting.

  *value* must be an integer from 1 through 65535. Default: 2 seconds.
- **commit**: Commits the configuration.

### Sample Configuration

The following is a sample configuration.

```
config
   profile radius
      timeout 4
      exit
```



CHAPTER 6

# Control Plane and User Plane Association



# Feature Summary and Revision History

## Summary Data

*Table 14: Summary Data*

| | |
|---|---|
| Applicable Product(s) or Functional Area | cnBNG |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Disabled - Configuration Required |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

## Revision History

*Table 15: Revision History*

| Revision Details | Release |
|---|---|
| First introduced. | 2021.01.0 |

# Feature Description

The Control Plane (CP) associates with a peer User Plane to synchronize with the number of subscriber sessions and state of each session. The CP and UP must maintain the total number of active sessions and their state on both sides.

To associate a UP to the CP, see the Associating the User Plane, on page 62.

# Enabling Control Plane and User Plane Association

This section describes how to enable CP to UP association.

Associating the CP and UP involves the following procedure.

Associating the User Plane

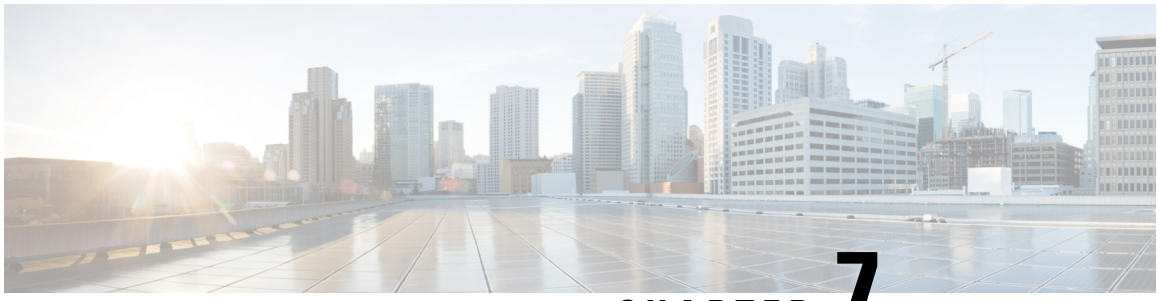
## Associating the User Plane

Use the following commands to associate the Control Plane (CP) to the peer User Plane.

```
config
   user-plane user_plane_name
   offline
   peer-address ipv4 ipv4_address
   port-id port_identifiersubscriber-profile subscriber_profile
   subscriber-profile subscriber_profile
   exit
```

**NOTES**:

- **user-plane** *user_plane_name*: Specifies the User Plane (UP) name and enter UP Configuration mode.
- **offline**: Marks the UP offline for a graceful disconnect.
- **peer-address ipv4** *ipv4_address*: Specifies the peer ipv4 address of the UP.
- **port-id** *port_identifier***subscriber-profile** *subscriber_profile*: Specifies the port identifier of the UP. **subscriber-profile** *subscriber_profile* associates the subscriber profile at the port identifier level.
- **subscriber-profile** *subscriber_profile*: Associates the subscriber profile at UP level.



CHAPTER 7

# DHCP and IPoE Subscriber Management



# Feature Summary and Revision History

## Summary Data

***Table 16: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | cnBNG |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Disabled - Configuration Required |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

## Revision History

***Table 17: Revision History***

# Feature Description

A session represents the logical connection between the customer premise equipment (CPE) and the network resource. To enable a subscriber to access the network resources, the network has to establish a session with the subscriber. The Cloud Native Broadband Network Gateway (cnBNG) supports the following subscriber session types:

- IPoE (DHCP)

- PPP (PPPoE)

  For more information, see PPPoE Subscriber Management , on page 113.

In an IPoE subscriber session, subscribers run IPv4 or IPv6 on the CPE device and connect to the BNG through a Layer-2 aggregation or Layer-3 routed network. The IP subscriber sessions that connect through a Layer-2 aggregation network are called L2-connected and sessions that connect through routed access network are called L3-connected or routed subscriber sessions. IPoE subscriber sessions are always terminated on BNG and then routed into the service provider network. IPoE relies on DHCP to assign the IP address.

On the BNG, the DHCPv4 or DHCPv6 trigger creation of these subscribers based on the First-Sign-Of-Life (FSOL) protocol. The IP sessions to the CPE can be either:

- Single stacked, that is, running only IPv4 or IPv6
- Dual stacked, that is, running both IPv4 and IPv6

The DHCP runs as a pod to handle the FSOL for the IPoE subscribers. It handles the DHCP packet encode and decode, IP address assignment, DHCP FSM handling, and DHCP feature and rule application for the IPoE sessions. The DHCP module handles both DHCPv4 and DHCPv6 control packets to bring up corresponding address family interface (AFI).

**Note** In this release, only the DHCP server mode functionality is supported.

A common DHCP module handles the DHCP finite state machines (FSM) for both 5G subscribers (in SMF service) and wireline subscribers in the cnBNG. The network function (NF) specific DHCP module handles the NF specific functionality.

# DHCP and IPoE Functionalities

The DHCP and IPoE Subscriber Management feature supports the following functionalities:

### DHCP Server

The cnBNG CP implementation supports the DHCPv4 server mode. The DHCP server FSM handles the DHCP packets from client, IP allocation, and IP lease management.

The FSM handles the following Rx control packets:

- Discover
- Request (DORA request and renew request)
- Decline
- Inform
- Release

The DHCP server FSM sends the following control packets to the client based on the FSM states and events:

- Offer
- Ack (DORA Ack, Renew Ack and Inform Ack)

- Noack

The DHCP server implementation associates a DHCP profile to a group of subscribers. This server implementation supports the following functionalities:

- IP address allocation for the client from the configured pool in the DHCP profile.
- IP address lease allocation based on DHCP profile configuration.
- Passing Host configurations to the client using the following configurable DHCP options in the DHCP profile:
  - IP subnet mask (Option 1)
  - Boot filename (Option 67)
  - Domain name (Option 15)
  - NetBIOS node type (Option 46)
  - NetBIOS name server (Option 44)
  - Domain name server (Option 6)
  - Default router (Option 3)
  - Time server (Option 4)

### Processing Option 82

cnBNG supports Option 82, which is the relay agent information option to figure out the sub-options. The various sub-options that the DHCP processes are:

- Circuit ID (Sub option 1)
- Remote ID (Sub option 2)

The circuit ID and remote ID field is passed to the Session Manager during session start trigger and the same is used for north-bound interactions.

### DHCPv4 RADIUS Proxy

The cnBNG CP supports DHCP IPv4 RADIUS proxy for RADIUS-based authorization of DHCP leases. This is a RADIUS-based address assignment mechanism in which a DHCP server authorizes remote clients and allocates IP addresses, based on replies from a RADIUS server.

These are the steps involved in the address assignment mechanism:

- The DHCP server sends the DHCP client information to the RADIUS server.
- The RADIUS server returns all required information, primarily IPV4 address, to the DHCP server in the form of RADIUS attributes. The subnet mask is derived from the CP based on the static pool configuration. The IPv4 address sent from the RADIUS must be part of the static pool associated to the UP.
- The DHCP server translates the RADIUS attributes into DHCP options and sends this information back in a DHCP Offer message to the DHCP client.

If the IETF attribute, such as Framed-IP-Address is received from the RADIUS server, and if it is present in the user profile, then this attribute is used instead of allocating the IP address from the configured pool. The basic attributes that can come from the RADIUS server that are relevant for DHCP server options are:

- Framed IPv4 Address
- IPv4 Subnet Mask (derived in the CP from the static pool configuration)
- IPv4 Default gateway (derived in the CP from the static pool configuration)

Apart from these attributes, the dhcp-class name and address pool name attribute also can come from RADIUS. If the RADIUS sets the address pool name, then it uses this for IP allocation instead of the pool that is specified as part of the DHCP profile.

If the RADIUS server sends the dhcp-class attribute to the DHCP server, then that attribute value is used to decide other configuration parameters in the reply that is to be sent to the DHCP client. For example, if the DHCPv4 server profile has both Class A and Class B in it, and if RADIUS server sends a reply to the DHCP server with the class name as 'B', then Class B is used to send the options back to the DHCP client. Classes can be defined under DHCP profile. The parameters and options that can be configured under DHCP profile can be configured under class also.

Additional RADIUS server attributes are allowed, but not mandatory. If a RADIUS server user profile contains a required attribute that is empty and is not available via configuration as well, the DHCP server does not generate the DHCP options.

### DHCPv6 Local Server for IPv6 Subscribers

The DHCPv6 server assigns IPv6 address and prefix and other configuration attributes (such as domain name, the domain name server address and SIP servers and so on) to requesting clients. On receiving a valid request, the server assigns the client IPv6 address or prefix, a lease for the assigned IPv6 address or prefix and other requested configuration parameters. The DHCP server FSM is implemented to handle the address allocation and lease management. The FSM would handle the following control packets from the client:

- Solicit
- Request
- Renew
- Rebind
- Decline
- Information-Request
- Release

The DHCPv6 server FSM sends the following control packets to the client based on the FSM states and events:

- Advertisement
- Reply (SARR Reply, Release Reply, Renew Reply, Rebind Reply and Information request Reply)
- Relay-Reply

The DHCPv6 server implementation associates a DHCPv6 profile to a group of subscribers. The server implementation caters to the following functionalities:

- IANA address and IAPD address allocation for the client from configured pool in DHCPv6 profile.
- IANA and IAPD address lease allocation based on DHCPv6 profile configuration.
- Passing Host configurations to client using below configurable DHCP options in DHCP profile
  - AFTR support (Option 64)
  - Preference option (Option 7)
  - Domain list (Option 24)
  - DNS server IPv6 address (Option 23)

The DHCPv6 server sends the following options to the Policy plane:

- interface-id (DHCP Option 18)
- remote-id (DHCP Option 37)
- vendor-class (DHCP Option 16)
- user-class (DHCP Option 15)
- client-id(DHCP Options 1)

### DHCPv6 Server - Prefix Delegation

The DHCPv6 Prefix Delegation feature enables the DHCPv6 server to hand out network address prefixes to the requesting clients. The clients use these network prefixes to assign /128 addresses to the hosts on their network. The RFC-3633 and RFC-3769 is supported for prefix delegation. The DHCPv6 Prefix Delegation feature is enabled by default for cnBNG DHCPv6 server. No other configuration is required to enable the prefix delegation. The DHCPv6 option OPTION_IA_PD (25) and OPTION_IAPREFIX (26) support to meet the prefix delegation requirement.

**Note**

- Only one delegated prefix per subscriber and client is supported.
- Only one OPTION_IAPREFIX is supported under one OPTION_IA_PD (25).

The cnBNG allocates addresses from the prefix pool configured under the DHCP profile.

### DHCPv6 Server - Address Assignment

The DHCPv6 Address Assignment feature enables the DHCPv6 server to hand out /128 addresses to the clients. The cnBNG DHCPv6 server implementation supports the DHCPv6 OPTION_IA_NA(3) and OPTION IAADDR(5) to enable address assignment to the client.

**Note**

- Only one delegated prefix per subscriber and client is supported.
- Only one OPTION_IAPREFIX is supported under one OPTION_IA_PD (25).

The cnBNG allocates addresses from the prefix pool configured under the DHCP profile.

### Prefix and Address Pool Support for IPv6

The cnBNG supports the configuring of the DHCPv6 address and prefix pool and associating it to the DHCPv4 and DHCPv6 server profiles. The address and prefix ranges is under the pool. cnBNG also supports downloading of the address and prefix pool name via the user profile on a per subscriber basis. The pool name downloaded via user profile is given priority over the pool name association via the DHCPv6 profile.

### DHCPv6 Server with RADIUS-based Address Assignment

The cnBNG supports RADIUS-based address assignment, that is, the IANA address is downloaded as part of the user profile and is allocated to the client. Address from the user profile is given priority over the local configuration.

### DHCPv6 Server with RADIUS-based Prefix Delegation

The cnBNG supports RADIUS-based prefix assignment, that is, the IAPD address is downloaded as part of the user profile and is allocated to the client. The delegated prefix from the user profile is given priority over the local configuration.

### DHCPv6-provided IPv6 address of DNS server for IPv6 Subscribers

The cnBNG CP DHCPv6 server implementation supports the provision of DNS server information to clients via the DNS option (23). It supports a configuration of up to 8 DNS server ipv6 addressees via the DHCPv6 profile. The DHCPv6 server information is downloaded via the user profile on a per subscriber basis. The per subscriber DNS information in the user profile is given priority over the profile configuration.

### DHCPv4 DHCPv6 Lease Timeout

The cnBNG CP provides the configuration to set the lease value under the DHCPv4 and DHCPv6 profile. This configuration determines the lease for the IP addresses allocated to the clients.

For DHCPv4 clients, the lease is set in the address time (T) option (option 51). By default, the renewal time is set as (½) * T [option 58] and rebinding time is set as (7/8) * T [option 59]. For DHCPv6 client, the lease is populated in the IA address and IA prefix option for the respective address types. By default, preferred time is set as 0.5 * T and valid time T2 is set as 0.8 * T. By default, renewal time (T1) is set as 0.5 * T and rebinding time T2 is set as 0.8 * T in OPTION_IA_PD.

The cnBNG CP tracks the lease time allocated to the clients. Ideally the client should renew (Renew request) the lease at T1 to extend the lease. If renew is failing, the client uses the rebind (broadcast request message for DHCPv4 and rebind message for DHCPv6). If the cnBNG CP does not receive the lease renewal request from the client, the lease times out after T and the corresponding address is released to the pool and removed from the client session. This can lead to an update or disconnect to the Session Manager based on the other address states. The lease timeout is applicable to both IPv4 and IPv6 addresses.

### IPv6 IPoE Sessions

The IPv6 subscribers run the IPv6 from the CPE device to the BNG router and are created using the DHCPv6 protocol. The IPv6 subscribers natively run IPv6 on the CPE device and are connected to the router via a Layer-2 network or through Layer-2 aggregation.

The IPv6 subscribers are supported when they are directly connect to the cnBNG UP or via a Layer-2 aggregator. The cnBNG CP DHCPv6 server treats only DHCPv6 SOLICIT message from the subscriber / client as FSOL (First Sign Of Life) packet in case of IPoE and initiates the subscriber session creation.

**Note** Routed subscribers are not supported.

### Dual Stack IPv6/IPv4 over IPoE

The cnBNG CP supports dual-stack IPoE subscribers, that is, both IPv4 and IPv6 address allocation for the same subscriber. In this release, cnBNG supports up to one IPv4 address, one IANA address, and one IAPD address.

### Subscriber Termination over Non-default VRF

The cnBNG CP DHCPv4 and DHCPv6 servers are VRF aware. The DHCPv4 and DHCPv6 servers support the access interface in either default VRF or non-default VRF. The following table shows the VRF combination supported by DHCPv6 server.

***Table 18: DHCP Supported VRF Combinations***

| Client Access Interface | Subscriber Interface | DHCPv6 Supported |
|---|---|---|
| Default VRF | Default VRF | Supported |
| Default VRF | Non-default VRF | Supported |
| Non-default VRF | Non-default VRF | Supported |

### DHCPv4 Raw Option Support

The cnBNG DHCP Profile configuration enables the operator to configure specific DHCPv4 options, under the DHCPv4 profile. The option value can range from 1 to 255. The option value can be either an ascii string or a hexadecimal string.

### DHCPv4 and DHCPv6 Class Support

The cnBNG DHCP Profile configuration enables the operator to configure classes of DHCP options and to selectively associate them during the session setup. The DHCP Options class are selected based on certain matching DHCP options received from access network against the configured class key parameters. The DHCP Options class can also be selected based on the class name received from Policy plane. The priority is always given to the DHCP class name that the Policy plane provides. However, if the Policy plane does not provide a class name, then class selection depends on the operator-configured key parameters. The operator can configure multiple DHCP option classes for DHCPv4 and DHCPv6 separately.

The DHCP Profile consist of profile elements. Each of the DHCPv4 and DHCPv6 profiles contain the ‘default’ DHCP options list and zero or more classes of DHCP options of corresponding DHCP version.

The DHCPv4 and DHCPv6 Options Class contains a list of DHCP options and the “Match-Info” holds the information about the keys to be matched to select that class. The operator can also specify under Match-Info” the class selection that should match ‘any’ or ‘all’ the key parameters of that class.

If the DHCP Option class does not match an ongoing session or any requested DHCP Options is not found in the selected class, then the requested option is selected from the 'default' DHCP Options of that profile.

# How it Works

This section provides a brief of how the DHCP and IPoE Subscriber Management feature works.

## Call Flows

This section includes the following call flow.

### cnBNG IPoE Call Flow

For IPoE session establishment, the BNG User Plane (UP) sends the DHCP packets to the BNG Control Plane (CP) using the GTP-U protocol. The following figure shows the DHCP packet call-flow and session programming between the BNG-UP and BNG-CP for IPoE session establishment.

***Figure 6: cnBNG IPoE Call Flow***

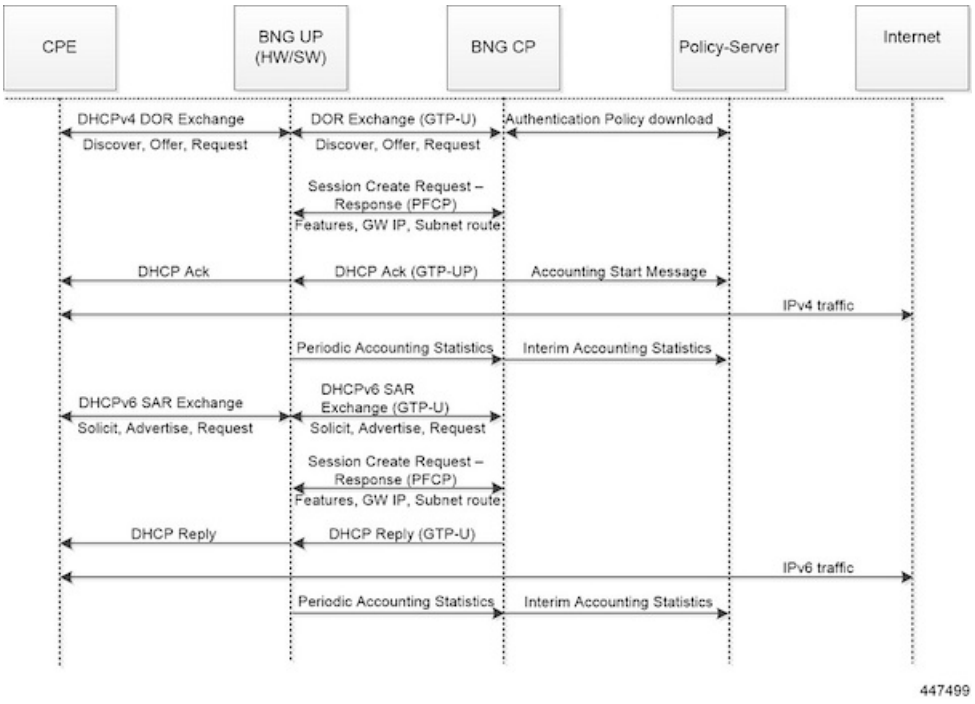



***Table 19: cnBNG IPoE Call Flow Description***

| Step | Description |
|---|---|
| 1 | The subscriber running IPv4 or IPv6 stack on the CPE device connects to the BNG-UP via DHCPv4, DHCv6, or DHCPv4 and DHCPv6. |
| 2 | The BNG-UP forwards the DHCP(v4/v6) request packets received from the CPE to the BNG-CP over the GTPU protocol. It then returns the DHCP response packets received from the BNG-CP to the CPE device. |

| Step | Description |
|---|---|
| 3 | The BNG-CP performs the subscriber authentication via the Policy Server before establishing a subscriber session on the BNG-UP. |
| 4 | After the BNG-CP successfully establishes a session on the BNG-UP, the BNG-UP initiates the Accounting Start and trigger Session Establishment Success (DHCPv4 Ack / DHCPv6 Reply) message towards the CPE via the BNG-UP. |
| 5 | The subscriber on the CPE device initiates the data traffic (DHCPv4 / DHCPv6) via the BNG-UP or BNG-CP towards the Internet. |
| 6 | The BNG-UP forwards the periodic accounting information to the BNG-CP and the BNG-CP triggers periodic accounting towards the Policy server. |

## Standard Compliance

The DHCP and IPoE Subscriber Management feature caters to the DHCP server requirements only. The DHCP Server implementation is aligned with the following standards:

- RFC 2131 Dynamic Host Configuration Protocol
- RFC 2132 DHCP Options and BOOTP Vendor Extensions [Subset of options]
- RFC 3046 DHCP Relay Agent Information Option
- RFC 3004 The User Class Option for DHCP
- RFC 3315 Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
- RFC 3633 IPv6 Prefix Options for Dynamic Host Configuration Protocol(DHCP)version 6
- RFC 3646 DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
- RFC 4649 Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option
- RFC 6334 Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite

## Limitations and Restrictions

The DHCP and IPoE Subscriber Management feature has the following limitations in this release:

- Only Layer 2 connected subscribers are supported.
- DHCPv6 addresses and prefixes do not get released at IPv6CP disconnect.
- For DHCPv4 sessions, subnet mask and default gateway are derived from the IPAM pool configuration and IP pool split logic. The first subnet route, subnet mask, and default gateway IP is derived from the IPAM and pushed to the UP for each chunk of the pool. Subnet mask and default gateway cannot be assigned via the AAA configuration.
- For DHCPv4 sessions, subnet selection is not supported. The IP is selected from the mapped IP pool. Subnet selection cannot be controlled via the AAA gateway IP, giaddr, or subnet selection suboption.
- For DHCPv4 sessions, requested IP option (option 50) that helps in requesting specific IP is not supported. However on client reboot (discover in bound state), the already assigned IP is retained.

- DHCP Inform packet and DHCPv6 Information Request packet handling for unbound sessions are not supported. That is, the client cannot get only the host configurations without requesting for IP assignment via BNG.
- For DHCPv6 session, multihop relay forward DHCPv6 message is not supported (as in physical BNG).
- For DHCPv4 session, broadcast flag check, and discovery, offer, request, and acknowledgement (DORA) unicast is not supported.
- If DHCP client initiated packet options like requested options (option 55 for IPv4, ORO option 6 for IPv6), circuit-id, remote-id, user class, vendor class changes in the packet over the session lifecycle, the cnBNG server behaviour is not defined. cnBNG assumes that the client will not change these options over the lifecycle of session. The client should also maintain the same values for attributes like remote-id, vendor class, user class for both IPv4 and IPv6 afi (AFI). In case these value are required to be changed, it is recommended to clear the session and bring it up again.
- Client reboot scenarios do not tear down the session in cnBNG in the following scenarios: If the Discover message is received in the Bound state or Solicit message is received for the already bound IANA, cnBNG does not tear down the existing session. Instead, the already allocated IP is assigned to the subscriber. In this case, fresh lease is assigned to the client. This is a difference in behaviour from physical BNG where on receiving Discover message in Bound state, IPv4 stack is brought down and new IP is assigned.
- No parity support for RADIUS attribute formatting with ASR 9000. The supported RADIUS attribute list and formatting would be updated based on feedback from customer. For example, some attributes like remote-id format is different for IPv4 and IPv6 clients. Hence, the value going to the Policy Plane differs based on whether the IPv4 or IPv6 afi comes up first.
- Change of Authorization (CoA) for DHCP consumed RADIUS attributes are not supported.
- RFC recommended DHCPv4/v6 packet validations are not supported.
- A common DHCP class attribute is used for class specification for DHCPv4 and DHCPv6 stack via AAA attribute. The attribute is dhcp-class.
- Framed route is not supported.
- Manual pod restart is not supported or entertained. Pod restart can lead to inconsistencies between the CP pods with regard to session count and session state. To recover the inconsistent sessions, the **clear** command must be used explicitly.
- After subscriber is up, if the subscriber is deleted from the cnBNG CP (for reasons like admin clear or Pod ) the subscriber is not notified. Therefore, the client must be explicitly rebooted for re-establishing the session. However, if the client is not rebooted explicitly, on receiving the Renew request. cnBNG ignores the renew request. Because the subscriber will retry till the lease expiry, renegotiation (with Discover and Solicit) occurs when the lease time is expired. Therefore, the subscriber loses connectivitiy till lease expiry (as session is already cleared in CP & UP) and explicit client reboot is required.

# Configuring the DHCP and IPoE Subscriber Management Feature

This section describes how to configure the DHCP and IPoE Subscriber Management feature.

Configuring the DHCP and IPoE Subscriber Management feature involves the following steps:

1. Configure the IPv4 DHCP Profile

2. Configure the IPv4 DHCP Class
3. Configure the IPv6 DHCP Profile
4. Configure the IPv6 DHCP Class

# Configuring the IPv4 DHCP Server Profile

Use the following commands to configure the IPv4 DHCP server profile.

```
config
  profile dhcp dhcp_profile_name
  ipv4
  server { boot-filename boot_filename } | { dns-servers dns_server } | {
domain-name domain_name } |
  { netbios-name-server netbios_name_server } | { netbios-node-type {
broadcast-node | hexadecimal | hybrid-node | mixed-node | peer-to-peer-node
 } |
  { next-server ipv4_address } | { ntp-servers ipv4_address } | { pool-name
ipam_pool_name } |
  { option-codes option_codes_range { ascii-string value | force insert { true
 | false } | hex-string value |
  { ip-address ip_address } | { lease { days value | hours value| minutes
value }
  exit
exit
```

**NOTES**:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv4**: Enters IPv4 configuration mode.
- **server { boot-filename** *boot_filename* **} | { dns-servers** *dns_server* **} | { domain-name** *domain_name* **} | { netbios-name-server** *netbios_name_server* **} | { netbios-node-type { broadcast-node | hexadecimal | hybrid-node | mixed-node | peer-to-peer-node } | { next-server** *ipv4_address* **} | { ntp-servers** *ipv4_address* **} | { pool-name** *ipam_pool_name* **} | { option-codes** *option_codes_range* **{ ascii-string** *value* **| force insert { true | false } | hex-string** *value* **| { ip-address** *ip_address* **} | { lease { days** *value* **| hours** *value*| **minutes** *value* **}**: Specifies the IPv4 server details.
  - **boot-filename** *boot_filename*: Configures the boot file.
  - **dns-servers** *dns_server*: Specifies the Domain Name System (DNS) IPv4 servers available to a DHCP for an IPv4 client.
  - **domain-name** *domain_name*: Specifies the domain name for the IPv4 client.
  - **netbios-name-server** *netbios_name_server*: Configures the NetBIOS name servers.
  - **netbios-node-type { broadcast-node | hexadecimal | hybrid-node | mixed-node | peer-to-peer-node }**: Configures the NetBIOS node as a broadcast, hexadecimal, hybrid, mixed, or peer-to-peer node. The valid values for each of these nodes are:
    - **broadcast-node**: 0x1 B-node
    - **hexadecimal**: Operator provided custom 1 byte hex value

  - **hybrid-node**: 0x8 H-node
  - **mixed-node**: 0x4 M-node
  - **peer-to-peer-node**: 0x2 P-node

- **next-server** *ipv4_address*: Specifies the TFTP-server IP address for the client to use.
- **pool-name** *ipam_pool_name*: Specifies the IP Address Management (IPAM) assigned pool name.
- **option-codes option_codes_range { ascii-string** *value* **| force insert { true | false } | hex-string** *value* **| ip-address** *ip_address* **}**: Specifies the values for the ASCII string of length 128, force insert, hex string of length 128, or IP address (IPv4 IP address).
- **lease { days** *value* **| hours** *value* **| minutes** *value* **}**: Specifies the lease time duration in the number of days, hours, and minutes. The number of lease days supported is from 0 to 365. The number of leave hours supported ranges from 0 to 23 and minutes from 0 to 59.

# Configuring the IPv4 DHCP Class

Use the following commands to configure the IPv4 DHCP class.

```
config
  profile dhcp dhcp_profile_name
  ipv4
  class dhcp_class_name
  matches { match { dhcpv4-circuit-id { ascii value | hex value } |
dhcpv4-remote-id { ascii value |
    hex value } | dhcpv4-vendor-class { ascii value | hex value } |
dhcpv4-user-class { ascii value |
    hex value } } | match-type { all match_key_value | any match_key_value } }
  end
```

**NOTES**:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv4**: Enters IPv4 configuration mode.
- **class** *dhcp_class_name*: Creates a proxy profile class (DHCP), which can be used to enter the proxy profile class sub-configuration mode.
- **matches { match { dhcpv4-circuit-id { ascii** *value* **| hex** *value* **} | dhcpv4-remote-id { ascii** *value* **| hex** *value* **} | dhcpv4-vendor-class { ascii** *value* **| hex** *value* **} | dhcpv4-user-class { ascii** *value* **| hex** *value* **} } | match-type { all** *match_key_value* **| any** *match_key_value* **} }**: Specifies the list of match keys and values. The match values supported are DHCPv4 circuit ID, DHCPv4 remote ID, DHCPv4 vendor class, and DHCPv4 user class. Each of the values must specify either an ASCII or hexadecimal value.

  **match-type { all | any }**: Specifies if the match value should apply to any of the specified keys or to all the keys.

# Configuring the IPv6 DHCP Server Profile

Use the following commands to configure the IPv6 DHCP server profile.

```
config
  profile dhcp dhcp_profile_name
    ipv6
      server { aftr-name value | dns-servers dns_server
      | domain-name domain_name | iana-pool-name ipam_pool_name
      | iapd-pool-name ipam_pool_name | lease { days value | hours value |
minutes value }
      | preference value }
```

**NOTES**:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv6**: Enters IPv6 configuration mode.
- **server { aftr-name** *value* **| dns-servers** *dns_server* **| domain-name** *domain_name* **| iana-pool-name** *ipam_pool_name* **| iapd-pool-name** *ipam_pool_name*| **lease { days** *value* **| hours** *value* **| minutes** *value* **} | preference** *value* **}**: Specifies the IPv6 server details.
  - **aftr-name** *value*: Specifies the FQDN string.
  - **dns-servers** *dns_server*: Specifies the Domain Name System (DNS) IPv4 servers available to a DHCP for an IPv4 client.
  - **domain-name** *domain_name*: Specifies the domain name for the IPv4 client.
  - **iana-pool-name** *ipam_pool_name*: Specifies the Internet Assigned Numbers Authority (IANA) pool name.
  - **iapd-pool-name** *ipam_pool_name*: Specifies the Identity Association for Prefix Delegation (IAPD) pool name.
  - **lease { days** *value* **| hours** *value* **| minutes** *value* **}**: Specifies the lease time duration in the number of days, hours, and minutes. The number of lease days supported is from 0 to 365. The number of leave hours supported ranges from 0 to 23 and minutes from 0 to 59.
  - **preference** *value*: Specifies the DHCP server preference. The preference value ranges from 1 to 255.

# Configuring the IPv6 DHCP Class

Use the following commands to configure the IPv6 DHCP class.

```
config
  profile dhcp dhcp_profile_name
    ipv6
      class dhcp_class_name
       server { aftr-name value | dns-servers dns_server | domain-name
domain_name |
       iana-pool-name ipam_pool_name | iapd-pool-name ipam_pool_name | lease {
 days value |
```

```
hours value | minutes value } preference value
end
```

**NOTES**:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv6**: Enters IPv6 configuration mode.
- **class** *dhcp_class_name*: Creates a proxy profile class (DHCP), which can be used to enter the proxy profile class sub-configuration mode.
- **server { aftr-name** *value* **| dns-servers** *dns_server* **| domain-name** *domain_name* **| iana-pool-name** *ipam_pool_name* **| iapd-pool-name** *ipam_pool_name* **| lease { days** *value* **| hours** *value* **| minutes** *value* **} | preference** *value* **}**: Specifies the IPv6 class server details.
  - **aftr-name** *value*: Specifies the FQDN string.
  - **dns-servers** *dns_server*: Specifies the Domain Name System (DNS) IPv6 servers available to a DHCP for an IPv6client.
  - **domain-name** *domain_name*: Specifies the domain name for the IPv6 client.
  - **iana-pool-name** *ipam_pool_name*: Specifies the Internet Assigned Numbers Authority (IANA) pool name.
  - **iapd-pool-name** *ipam_pool_name*: Specifies the Identity Association for Prefix Delegation (IAPD) pool name.
  - **lease { days** *value* **| hours** *value* **| minutes** *value* **}**: Specifies the lease time duration in the number of days, hours, and minutes. The number of lease days supported is from 0 to 365. The number of leave hours supported ranges from 0 to 23 and minutes from 0 to 59.
  - **preference** *value*: Specifies the DHCP server preference. The preference value ranges from 1 to 255.

# DHCP IP Lease Reservation

## Feature Summary

***Table 20: Feature Summary***

| | |
|---|---|
| Applicable Product(s) or Functional Area | cnBNG |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Disabled - Configuration Required |
| Related Documentation | Not Applicable |

# Revision History

*Table 21: Revision History*

| Revision Details | Release |
|---|---|
| First introduced | 2022.04.0 |

# Feature Description

DHCP IP Lease Reservation feature enables the DHCP to allocate an IP address dynamically when the subscriber logs into the network the first time. Then, the assigned IP address can be reserved permanently for the subscriber, which means, the same IP address is assigned every time the subscriber logs in.

# How it Works

This section provides a brief of how the DHCP IP Lease Reservation feature works.

After the DHCP IP Lease Reservation feature is enabled (see Configuring DHCP IP Lease Reservation, on page 77), if a subscriber (CPE) logs into the system for the first time, IPAM allocates an IP address dynamically from the IP pool. Administrators can use the REST API/action command (see Reserving IP Address using CLI (Action Command/REST API), on page 78) to reserve the IP address for the subscriber. So, when the same session is initiated the next time, the DHCP provides the same IP address to the subscriber.

**Note** If you do not want to reserve the IP address, the administrators can use the same REST API/action command with **delete** option and clear the IP lease reservation.

# Limitations and Restrictions

The DHCP IP Lease Reservation feature has the following limitation:

- The DHCP IP Lease Reservation and Leased IP Hold Time features cannot be used together at the same time.

# Configuring DHCP IP Lease Reservation

Use the following commands to enable/disable the DHCP IP Lease Reservation feature:

```
config
   [ no ] subscriber featurette dhcp-lease-reservation enable
end
```

**NOTES:**

- **subscriber featurette dhcp-lease-reservation enable**: Enables the DHCP IP Lease Reservation feature
- **no subscriber featurette dhcp-lease-reservation enable**: Disables the DHCP IP Lease Reservation feature

# Reserving IP Address using CLI (Action Command/REST API)

Administrators can use the following action command/REST API to reserve the addresses (IPv4, IANA, and IAPD) that are allocated to the subscriber with a specific username.

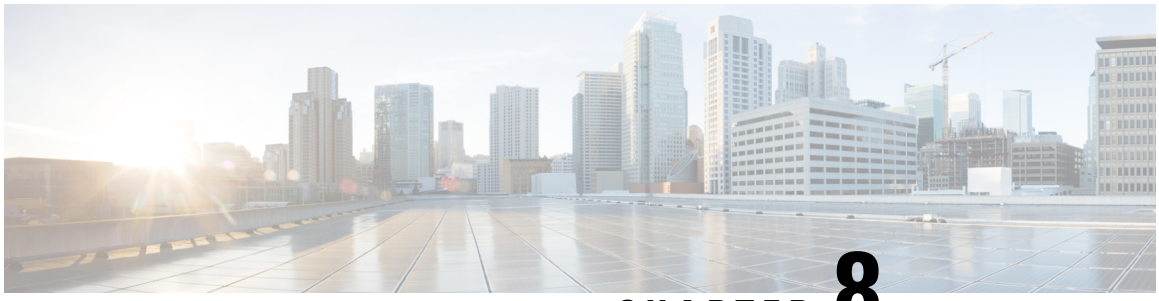
```
bng# subscriber lease-reservation subkey username_string [ delete ]
```

**NOTES**:

- **subkey** *username_string* : Specifies the username for which the IP addresses are reserved.
- **delete**: Clears the lease reservation for the specific username.

**Note**

- This command/REST API fails if the subscriber is disconnected.
- This command/REST API fails if the DHCP IP Lease Reservation feature is not enabled.



CHAPTER 8

# IP Address Management



# Feature Summary and Revision History

## Summary Data

***Table 22: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | cnBNG |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Disabled - Configuration Required |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

## Revision History

***Table 23: Revision History***

| Revision Details | Release |
|---|---|
| First introduced. | 2021.01.0 |

# Feature Description

IP Address Management (IPAM) is a method of tracking and managing IP addresses of a network. IPAM is one of the core components of the subscriber management system. Traditional IPAM functionalities are insufficient in Cloud-Native network deployments. Hence, IPAM requires additional functionalities to work

with the Cloud-Native subscriber management system. The Cloud-Native IPAM system is used in various network functions, such as Session Management function (SMF), Policy Charging function (PCF), and Broadband Network Gateway (BNG).

The IPAM system includes the following functionalities to serve the Cloud Native and Control and User Plane Separation (CUPS) architecture:

- **Centralized IP Resource Management**—Based on the needs of the Internet Service Provider (ISP), the Control Plane (CP) is deployed either on a single (centralized) cluster or multiple (distributed) clusters. For multiple cluster deployments, the IPAM automatically manages the single IP address space across the multiple CPs that are deployed in the distributed environment.
- **IP Address-Range Reservation per User Plane**—For subscribers connecting to the Internet core, the User Plane (UP) provides the physical connectivity. The UP uses the summary-routes to advertise subscriber routes to the Internet core. For CPs that are managing multiple UPs, the CP reserves a converged IP subnet to the UPs. In such a scenario, the IPAM splits the available address space into smaller address-ranges and assigns it to different UPs.
- **IP Address Assignment from Pre-Reserved Address-Ranges**—When subscribers request for an IP address, the IPAM assigns addresses from the pre-reserved address range of their respective UP.

# IPAM Components

This section describes the different components of the IPAM system.

## IPAM Sub-Modules

The IPAM functionalities are categorized in the following sub-modules:

### IPAM Server

This module manages the complete list of pools and address-space configurations. It splits the configured address-ranges into smaller address-ranges (statically or dynamically) to distribute it to the IPAM Cache modules. The IPAM server can be deployed as a centralized entity to serve a group of CN clusters or as an integrated entity within a single cluster.

### IPAM Cache

This module acquires the free address-ranges from the IPAM server and allocates individual IP addresses to the IPAM clients. The IPAM cache is generally deployed in the Distributed mode running within each cluster, to communicate with the co-located or remotely located IPAM server. It is also responsible for address-range reservation per UP and pool threshold monitoring. The IPAM server and cache modules can also run in an integrated mode.

### IPAM Client

This module is tightly coupled with its respective network-function, responsible for handling request and release of individual IP address from the IPAM cache for each IP managed end-device.

Unlike the IPAM server and cache module, the IPAM client caters to use-cases specific to network-functions such as BNG, SMF, PCF, and so on.

# IPAM Integration in cnBNG

The Cloud-Native Broadband Network Gateway (cnBNG) function comprises of loosely coupled microservices that provide the functionality of the BNG. The decomposition of these microservices is based on the following three-layered architecture:

1. Layer 1: Protocol and Load Balancer Services (Stateless)
2. Layer 2: Application services (Stateless)
3. Layer 3: Database Services (Stateful)

The IPAM and cnBNG integration occurs in the Application Services layer.

**BNG Node Manager Application**—The BNG Node Manager application is responsible for the User Plane function (UPF) management, ID and resource management, and IP address management. Therefore, the IPAM Cache is integrated as part of this microservice.

Also, the UPF uses the IPAM Client module for address-range-reservation per UPF.

**BNG DHCP and PPPOE Application**—The BNG-DHCP and BNG-PPOE pods are responsible for providing IP addresses to the BNG subscriber session. During session bring-up, the IP address is requested and during session bring-down, the IP address is released back. These First Sign of Life (FSOL) applications send the inter-process communications (IPC) to the Resource Manager (RMGR) component in the NodeMgr. The NodeMgr receives the IPC and invokes the IPAM component.

**IPAM Server Application**—Based on the deployment model, the IPAM Server runs as an independent microservice as part of the same cluster or in a remote cluster.

In standalone deployments, the IPAM Server functionality is an integral part of the IPAM Cache, that is, it runs as part of the Node Manager microservice itself.

# How it Works

This section describes the call flow pertaining to the integration of the IPAM in the cnBNG.

## Call Flows

This section describes the following IPAM call flows in cnBNG:

- IPAM initial sequence call flow
- IPAM call flow
- IPAM static-pool call flow

### IPAM Initial Sequence Call Flow

This section describes the cnBNG initial sequence call-flow.

*Figure 7: IPAM Initial Sequence Call Flow*

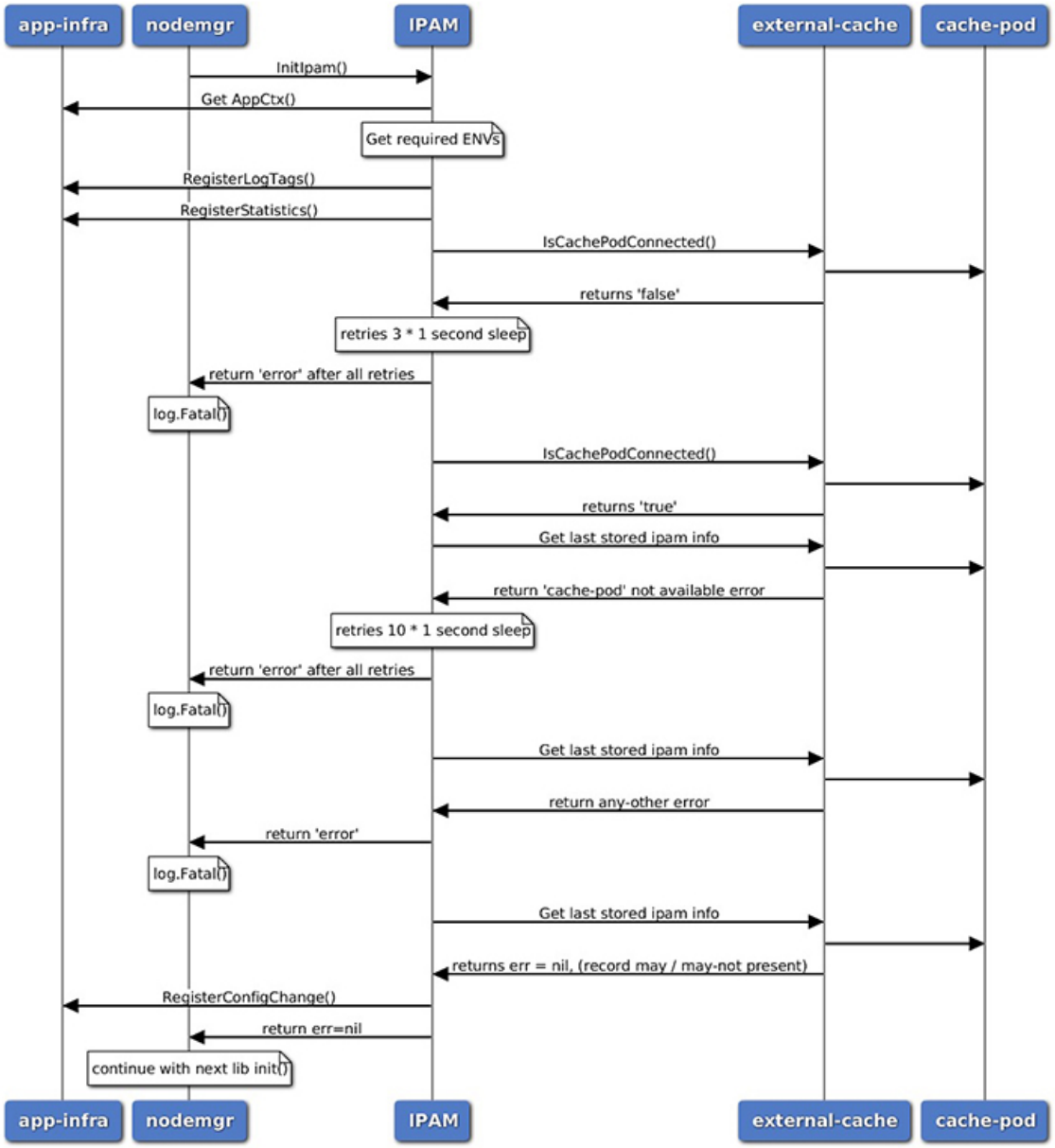



*Table 24: IPAM Initial Sequence Call Flow Description*

| Step | Description |
|---|---|
| 1 | IPAM reads the required environments, registers with the application infrastructure for log-tags, metrics, and database connection. |
| 2 | IPAM restores the previous state from the cache-pod, if present. |
| 3 | IPAM registers for configuration change and applies the new configuration change, if any. -change, apply new config-changes if any |

## IPAM Call Flow

This section describes the cnBNG IPAM call-flow.

*Figure 8: IPAM Call Flow*

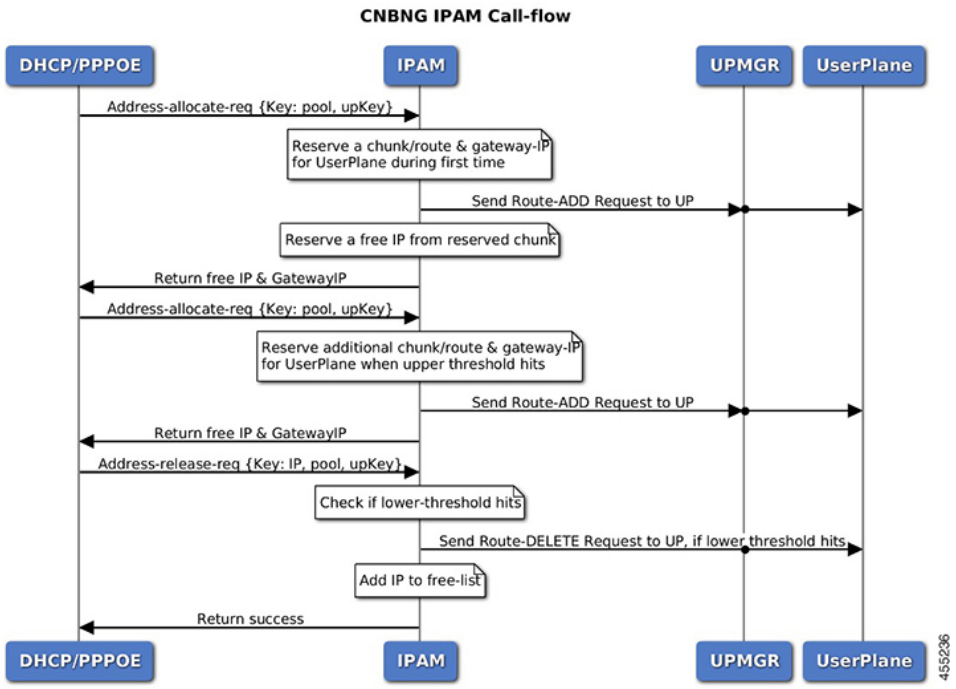



*Table 25: IPAM Call Flow Description*

| Step | Description |
|---|---|
| 1 | IPAM receives the ‘addr-alloc’ request from the DHCP or PPPoE pod with pool-name, addr-type and user plane function (UPF) as input. |
| 2 | IPAM reserves a new address-range (if not already present for UPF) and sends a ROUTE-ADD message to the UPF. It waits for a success or failure response. If the receives a failure response, it removes the chunk and repeats this step. |
| 3 | IPAM reserves a free-IP from the assigned address-range and returns to the DHCP or PPPoE. |
| 4 | IPAM monitors the ‘upper-threshold’ for each UPF during each IP address-allocation and also has a background thread that monitors. It then assigns new address-ranges to the UPF and repeats the ROUTE-ADD flow. |
| 5 | IPAM receives the ‘addr-free’ request from the DHCP or PPPoE pod with pool-name, addr-type, addr or pfx, and UPF as input. |
| 6 | IPAM moves the addr or pfx first to the quarantine-list until the quarantine timer and later moves it to the free-list. |

| Step | Description |
|---|---|
| 7 | IPAM monitors the ‘lower-threshold’ (currently 0%) of the address-range of each UPF, removes the address-range from the UPF, and sends the ROUTE-DELETE message. |

## IPAM Static-Pool Call Flow

This section describes the IPAM static-pool call flow.

*Figure 9: IPAM Static Pool Call Flow*

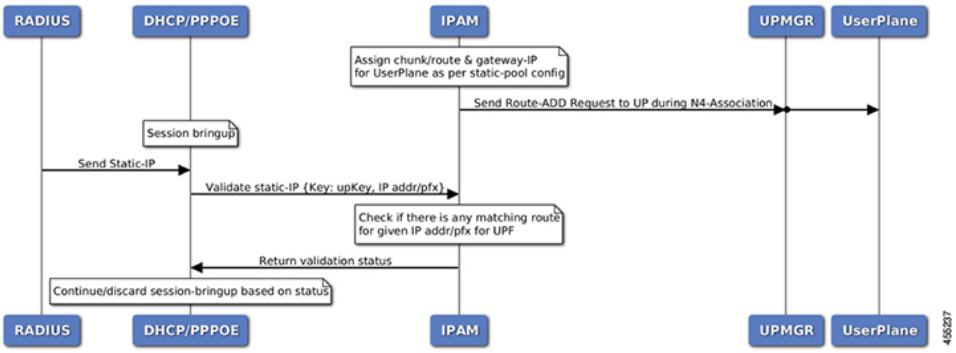



*Table 26: IPAM Call Flow Description*

| Step | Description |
|---|---|
| 1 | IPAM receives the ‘addr-alloc’ request from the DHCP or PPPoE pod with pool-name, addr-type and user plane function (UPF) as input. |
| 2 | IPAM reserves a new address-range (if not already present for UPF) and sends a ROUTE-ADD message to the UPF. It waits for a success or failure response. If the receives a failure response, it removes the chunk and repeats this step. |
| 3 | IPAM reserves a free-IP from the assigned address-range and returns to the DHCP or PPPoE. |
| 4 | IPAM monitors the ‘upper-threshold’ for each UPF during each IP address-allocation and also has a background thread that monitors. It then assigns new address-ranges to the UPF and repeats the ROUTE-ADD flow. |
| 5 | IPAM receives the ‘addr-free’ request from the DHCP or PPPoE pod with pool-name, addr-type, addr or pfx, and UPF as input. |
| 6 | IPAM moves the addr or pfx first to the quarantine-list until the quarantine timer and later moves it to the free-list. |
| 7 | IPAM monitors the ‘lower-threshold’ (currently 0%) of the address-range of each UPF, removes the address-range from the UPF, and sends the ROUTE-DELETE message. |

## Limitations

The IPAM feature has the following limitations:

- Duplicate IP address is not supported within a pool.
- Duplicate IP address is not supported across pools, that belong to same VRF.
- Removal of ‘pool’ is not supported while addresses are already assigned.
- Removal or modification of IP-address-ranges is not supported while addresses are already assigned.
- Change of ‘source’ field is not supported while address or prefixes are already assigned.
- Change of ‘vrf-name’ of pool is not supported while address or prefixes are already assigned.
- Start-address should be less than the End-address.
- Configuring addr-range split-size in wrong manner, that is, size of address-range < size-of-per-cache < size-of-dp, is not supported.
- Configuring IPv6 Address (IANA) and Prefix (IAPD) values interchangeably is not supported.
- Configuring invalid ‘prefix-length’ for Prefix (IAPD) range is not supported.

# Configuring IPAM Feature

This section describes how to configure the IPAM feature.

Configuring the IPAM feature involves the following steps:

1. Configuring IPAM source
2. Configuring the global threshold
3. Configure IPAM address pool
4. Configuring IPv4 address ranges
5. Configuring IPv6 address ranges
6. Configuring IPv6 prefix ranges
7. Configuring the IPv4 threshold
8. Configuring the IPv6 threshold
9. Configuring IPv4 address range split
10. Configuring IPv6 address and prefix address-range split

## Configuring IPAM Source

Use the following configuration to configure the IPAM source.

```
config
  ipam
```

```
    source local
      threshold { ipv4-add percentage | ipv6-address percentage | ipv6-prefix
percentage }
       commit
```

**NOTES**:

- **ipam**: Enters the IPAM Configuration mode.
- **source local**: Enters the local datastore as the pool source.
- **threshold { ipv4-add** *percentage* **| ipv4-address** *percentage* **| ipv6-prefix** *percentage* **}**: Specifies the threshold in percentage for the following:
  - **ipv4-add** *percentage*: Specifies the IPv4 threshold. The valid values range from 1 to 100. The default value is 80.
  - **ipv6-add** *percentage*: Specifies the IPv4 threshold. The valid values range from 1 to 100. The default value is 80.
  - **ipv6-prefix** *percentage*: Specifies the IPv6 threshold prefix. The valid values range from 1 to 100. The default value is 80.

# Configuring Global Threshold

Use the following configuration to configure the global threshold.

```
config
   ipam
      threshold
         ipv4-addr percentage
         ipv6-addr percentage
         ipv6-prefix percentage
         commit
```

**NOTES**:

- **ipam**: Enters the IPAM Configuration mode.
- **threshold**: Enters the threshold sub-mode.
- **ipv4-add** *percentage*: Specifies the IPv4 threshold. The valid values range from 1 to 100. The default value is 80.
- **ipv6-add** *percentage*: Specifies the IPv4 threshold. The valid values range from 1 to 100. The default value is 80.
- **ipv6-prefix** *percentage*: Specifies the IPv6 threshold prefix. The valid values range from 1 to 100. The default value is 80.

# Configuring IPAM Address Pool

Use the following configuration to configure the IPAM address pool.

```
config
  ipam
    address-pool pool_name [ address-quarantine-timer ] [offline ] [ static
 user_plane_name ] [ vrf-name string ]
     commit
```

**NOTES**:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool** *pool_name* **[ address-quarantine-timer ] [offline ] [ static** *user_plane_name* **] [ vrf-name** *string* **]**: Configures the address pool configuration. *pool_name* must be the name of the address pool.

  This command configures the following parameters:

  - **offline**: Sets the address pool to offline mode.
  - **static** *user_plane_name*: Specifies the 'user-plane' name associated to this static-pool.
  - **vrf-name** *string*: Configures the Virtual routing and forwarding (VRF) name of the pool.

# Configuring IPv4 Address Ranges

Use the following configuration to configure the IPv4 address ranges.

```
config
  ipam
     address-pool pool_name
        ipv4
           address-range start_ipv4_address end_ipv4_address [ default-gateway
ipv4_address ] [ offline ]
           commit
```

**NOTES**:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv4**: Enters the IPv4 mode of the pool.
- **address-range** *start_ipv4_address end_ipv4_address* **[ default-gateway** *ipv4_address* **] [ offline ]**: Configures the IPv4 address range with the starting and ending IPv4 address.
  - **default-gateway** *ipv4_address*: Specifies the IPv4 address of the default gateway.
  - **offline**: Sets the address pool to offline mode.

# Configuring IPv6 Address Ranges

Use the following configuration to configure the IPv6 address ranges:

```
config
  ipam
```

```
address-pool pool_name
   ipv6
      address-range start_ipv6_address end_ipv6_address [ offline ]
      commit
```

**NOTES**:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv6**: Enters the IPv6 mode of the pool.
- **address-range** *start_ipv6_address end_ipv6_address* **[ offline ]**: Configures the IPv6 address range with the starting and ending IPv6 address.

  **[ offline ]**: Sets the address pool to offline mode.

# Configuring IPv6 Prefix Ranges

Use the following configuration to configure the IPv6 prefix ranges:

```
config
   ipam
      address-pool pool_name
         ipv6
            prefix-ranges
               prefix-range prefix_value prefix-length prefix_length
               commit
```

**NOTES**:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool***pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv6**: Enters the IPv6 mode of the pool.
- **prefix-ranges**: Enters the prefix ranges mode.
- **prefix-range** *prefix_value* **prefix-length** *length*: Configures the IPv6 prefix range. *prefix_value* specifies the IPv6 prefix range.

  **prefix-length** *length* specifies the IPv6 prefix length.

# Configuring IPv4 Threshold

Use the following configuration to configure the IPv4 threshold:

```
config
   ipam
      address-pool  pool_name
         ipv4
            threshold
```

```
            upper-threshold  percentage
            commit
```

**NOTES**:

- **ipam**: Enters the IPAM Configuration mode.
- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv4**: Enters the IPv4 mode of the pool.
- **threshold**: Enters the threshold sub-mode.
- **upper-threshold** *percentage*: Specifies the IPv4 upper threshold value in percentage. The valid values range from 1 to 100. The default value is 80.

The following is a sample configuration:

```
config
   ipam
      address-pool p1
         ipv4
            threshold
               upper-threshold 80
```

# Configuring IPv6 Prefix-Range Threshold

Use the following configuration to configure the IPv6 prefix-range threshold.

```
config
   ipam
      address-pool pool_name
         ipv6
            prefix-ranges
               threshold
                  upper-threshold percentage
                  commit
```

**NOTES**:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv6**: Enters the IPv6 mode of the pool.
- **prefix-ranges**: Enters the IPv6 prefix ranges sub-mode.
- **threshold**: Enters the threshold sub-mode.
- **upper-threshold** *percentage*: Specifies the IPv6 upper-threshold value in percentage.

The following is an example configuration:

```
config
   ipam
      address-pool p3
         ipv6
```

```
prefix-ranges
   threshold
      upper-threshold 78
```

# Configuring IPv4 Address Range Split

Use the following configuration to configure the IPv4 address range split.

```
config
  ipam
    address-pool pool_name
      ipv4
        [ no ] split-size { per-cache value | per-dp value }
        commit
```

**NOTES**:

- **ipam**: Enters the IPAM configuration mode.
- -**address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv4**: Enters the IPv4 mode of the pool.
- **[ no ] split-size { per-cache** *value* **| per-dp** *value* **}**: Specifies the size of the IPv4 range to be split for each IPAM cache allocation. The IPAM server consumes this configuration. The **no** form of this command disables the splitting of the address-ranges into smaller chunks.

  **per-cache** *value*: Specifies the size of the IPv4 range to be split for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144. The default value is 1024.

  The IPAM cache consumes this configuration.
- **per-dp** *value*: Specifies the size of the IPv4 range to be split for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144 The default value is 256.

  The IPAM cache consumes this configuration.

# Configuring IPv6 Address and Prefix Address-Range-Spilt

Use the following configuration to configure the IPv6 address and prefix address range spilt.

```
config
  ipam
    address-pool pool_name
      ipv6
        address-ranges
          [ no ] spilt-size { per-cache value | per-dp value }
          commit
        prefix-ranges
          [ no ] spilt-size { per-cache value | per-dp value }
          commit
```

**NOTES**:

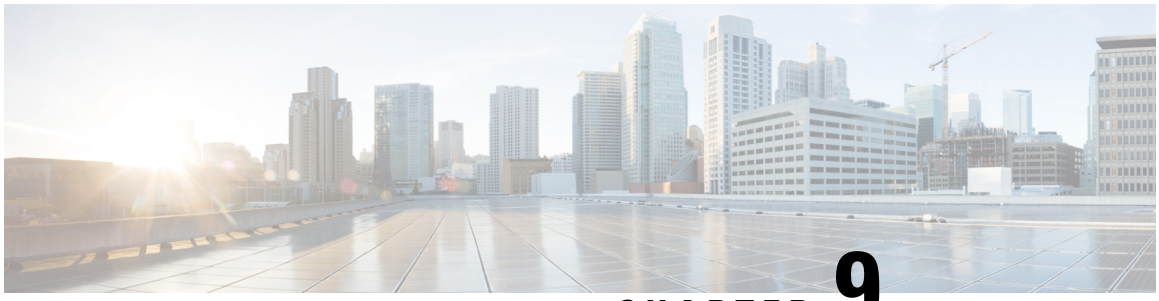
- **ipam**: Enters the IPAM configuration mode.

- **address-pool** *pool_name*: Configures the address pool. *pool_name* must be the name of the address pool.
- **ipv6**: Enters the IPv6 mode of the pool.
- **[ no ] spilt-size { per-cache** *value* **| per-dp** *value* **}**: Specifies the size of the IPv6 range to be split for each IPAM cache allocation. The IPAM server consumes this configuration. The **no** form of this command disables the splitting of the address-ranges into smaller chunks.

  **per-cache** *value*: Specifies the size of the IPv6 range to be spilt for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144. The default value is 1024.

  The IPAM cache consumes this configuration.
- **per-dp** *value*: Specifies the size of the IPv6 range to be spilt for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144 The default value is 256.

  The IPAM cache consumes this configuration.



CHAPTER 9

# Log Generation Support



# Feature Summary and Revision History

## Summary Data

***Table 27: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | cnBNG |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Enabled -Always-on |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

## Revision History

***Table 28: Revision History***

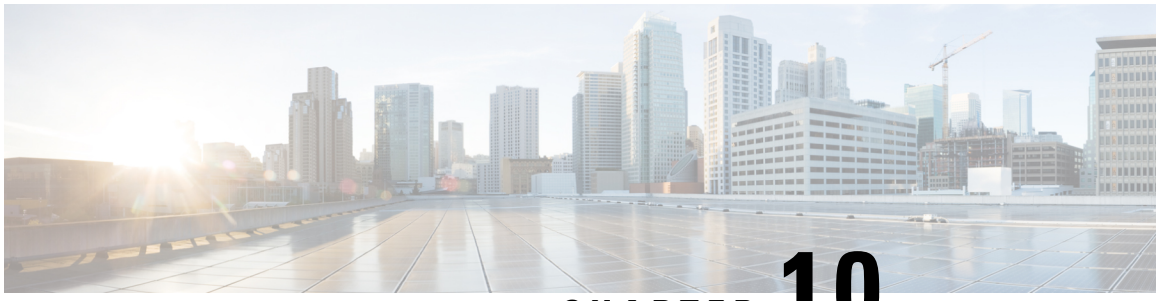
| Revision Details | Release |
|---|---|
| First introduced. | 2021.01.0 |

# Feature Description

The Cloud Native Broadband Network Gateway utilizes the common logging framework to generate logs from its microservices.

The supported logging levels are:

- Error
- Warn
- Info
- Debug
- Trace



CHAPTER 10

# Monitor Protocol and Subscriber



# Feature Summary and Revision History

## Summary Data

***Table 29: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | cnBNG |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Disabled - Configuration Required |
| Related Documentation | *Cloud Native BNG Control Plane Command Reference Guide* |

## Revision History

***Table 30: Revision History***

| Revision Details | Release |
|---|---|
| First introduced. | 2021.03.0 |

# Feature Description

The Monitor Subscriber and Protocol feature supports the debugging functionality.

### Monitor Subscriber

The Monitor Subscriber feature captures all the transactional logs for a given subscriber over a specified period of time across all the Kubernetes pods. It also supports the simultaneous monitoring of multiple subscribers on a given cluster. This information allows to track all the events that had occurred for a given subscriber when the subscriber was coming up or going down.

### Monitor Protocol

The Monitor Protocol feature replicates the packets from different protocol endpoints of cnBNG and sends it to the OAM pod. There two levels of packet replication that occur:

- First replication dumps only the basic packet information
- Second replication dumps the full packet with details like headers, keys of subscriber, and so on.

This feature captures all ingress and egress packets on the cnBNG protocol pods.

# Configuring Monitor Subscriber and Protocol

This section describes how to configure subscriber and protocol monitoring.

Configuring the Monitor Subscriber and Protocol feature involves the following procedures:

- Configuring Monitor Subscriber
- Configuring Monitor Protocol
- Copying Log Files
- Viewing Log Files

## Configuring Monitor Subscriber

Use the following commands to enable the monitoring of a subscriber.

**monitor subscriber supi** *subcriber_id* **capture-duration** *duration_in_seconds*

**NOTES**:

- **supi** *subcriber_id* : Enables monitoring of subscribers based on the subscriber identifier (supi). For example: 0000.4096.3e4a.

  The subscriber-id format supported is as follows:

  <mac-adress>@<upf>: This specifies a particular subscriber with the given MAC address from a specific User Plane function (UPF).

  Wildcard subscriber-id is also supported. For example:

  - *@<upf>: This specifies all subscribers from a specific UPF.
  - <mac>@*: This specfies all subscribers having the given MAC and from any UPF.
  - *: This specifies all subscribers from all UPFs.

- **capture-duration** : Specifies the duration in seconds during which the monitor subscriber is enabled. The *duration_in_seconds* can range from 1 to 2147483647 seconds. The default is 300.
- Other sub-options that are present in the CLI command are not supported

### Example

```
bng# monitor subscriber supi aabb.0000.0001@automation-userplane
supi: aabb.0000.0001@automation-userplane
captureDuraiton: 300
enableInternalMsg: false
enableTxnLog: false
namespace(deprecated. Use nf-service instead.): none
nf-service: none
gr-instance: 0
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   337  100   119  100   218  10818  19818 --:--:-- --:--:-- --:--:-- 30636
Command: --header Content-type:application/json --request POST --data
{"command":"mon_sub","params":{"supi":"aabb.0000.0001@automation-userplane","duration":300,"enableTxnLog":false,"enableInternalMsg":false,"action":"start","namespace":"none","nf-service":"none","grInstance":0}}
 http://oam-pod:8879/commands
Result start mon_sub, fileName
->logs/monsublogs/none.aabb.0000.0001@automation-userplane_TS_2021-06-09T12:17:33.838574118.txt
Starting to tail the monsub messages from file:
logs/monsublogs/none.aabb.0000.0001@automation-userplane_TS_2021-06-09T12:17:33.838574118.txt
Defaulting container name to oam-pod.
Use 'kubectl describe pod/oam-pod-0 -n bng' to see all of the containers in this pod.
      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.194843
      Message: BNGN4UdpProxyMessage
      Description: Received Packet IPOE, IPC Message from udp-proxy
      Source: bng.udp-proxy.DC.Local.0
      Destination: bng.bng-n4-protocol.DC.Local.0
      PAYLOAD:
        BNGN4UdpProxyMessage:
            BNGN4UdpProxyMessage:
                Type: 6
                L2Data:
                    SrcMac: aabb.0000.0001
                    DstMac: ffff.ffff.ffff
                    Outervlan: 100
                    Innervlan: 200
                    OuterCos: 0
                    InnerCos: 0
                IpAddr:
                    AfType: 1
                    SrcIpv4:
                    SrcIpv6:
                    DstIPv4: 8.8.8.8
                    DstIPv6:
                    LinkLocal:
                    Port: 8000
                UpData:
                    AccessInterface: GigabitEthernet0/0/0/1
                    CpSubscriberId: 0
                    UpSubscriberId: 0
                    UPSubInterfaceId: 0
                    RouterName: automation-userplane
                    AccessVrf: access-vrf-1
                    NASID: NAS-ID-1
                NasInfo:
                    Port: 4
```

```
                    Slot: 2
                    Adapter: 5
                    Subslot: 3
                    Chasis: 1
                    InterfaceType: 1
                L2TPData:
                    PuntPoliceRate: 0
                    L2TPTos: 0
                    TunnelID: 0
                Packet:
                    Payload:
                        BaseLayer:
                        Operation: 1
                        HardwareType: 1
                        HardwareLen: 6
                        HardwareOpts: 0
                        Xid: 1
                        Secs: 0
                        Flags: 32768
                        ClientIP: 0.0.0.0
                        YourClientIP: 0.0.0.0
                        NextServerIP: 0.0.0.0
                        RelayAgentIP: 0.0.0.0
                        ClientHWAddr: aa:bb:00:00:00:01
                        ServerName:
                        File:
                        Options: {
  Option(MessageType:Discover)
  Option(ClientID:[1 170 187 0 0 0 1])
}

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

      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.205174
      Message: RadiusUdpProxyMsg
      Description: Send Auth/Acct Request Message to UDP-Proxy
      Source: bng.radius-ep.DC.Local.0
      Destination: bng.udp-proxy.DC.Local.0
      PAYLOAD:
        RadiusUdpProxyMsg:
            RadiusUdpProxyMsg:
                SrcIp: 10.105.254.113
                SrcPort: 16384
                DestIp: 10.105.254.114
                DestPort: 1812
                Payload:
   Code = AccessRequest
   Id = 2
   Authenticator = [148 88 241 197 50 83 83 156 105 245 107 167 117 131 237 165]
   User-Name = "cnbng"
   User-Password = 0x30b19d11f96401290b6410e8a1b324eb
   NAS-IP-Address = 10.105.254.113
   NAS-Port = 16384
   Service-Type = 5
   Called-Station-Id = "1"
   Calling-Station-Id = "1"
   Nas-Identifier = "bng"
   Acct-Session-Id = "Local_DC_16777218"
   Event-Timestamp = 1623241161
   NAS-Port-Type = 41
   NAS-Port-Id = "124536"
   NAS-IPv6-Address = ::/0
   Cisco-Vsa_cisco-nas-port = "124536"
```

```
   Cisco-Vsa_cisco-dhcp-client-id = 0x01aabb00000001
   Cisco-Vsa_Cisco AVpair = "client-mac-address=aabb.0000.0001"
   Cisco-Vsa_Cisco AVpair = 0x646863702d636c69656e742d69643d01aabb00000001
               PayloadLen: 231
               SubscriberID: aabb.0000.0001@automation-userplane

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

      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.206778
      Message: RadiusUdpProxyMsg
      Description: Received Auth/Acct Response Message from UDP-Proxy
      Source: bng.udp-proxy.DC.Local.0
      Destination: bng.radius-ep.DC.Local.0
      PAYLOAD:
        RadiusUdpProxyMsg:
            RadiusUdpProxyMsg:
                SrcIp: 10.105.254.114
                SrcPort: 1812
                DestIp: 10.105.254.113
                DestPort: 16384
                Payload:
   Code = AccessAccept
   Id = 2
   Authenticator = [127 214 195 68 205 142 58 23 126 138 11 70 241 169 153 92]
               PayloadLen: 20

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

      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.216130
      Message: DHCPPacketTx
      Description: Sending Packet IPOE, IPC Message to udp-proxy
      Source: bng.bng-n4-protocol.DC.Local.0
      Destination: bng.udp-proxy.DC.Local.0
      PAYLOAD:
        DHCPPacketTx:
            DHCPPacketTx:
                Type: 6
                L2Data:
                    DstMac: ff:ff:ff:ff:ff:ff
                    Outervlan: 100
                    Innervlan: 200
                    OuterCos: 0
                    InnerCos: 0
                IpAddr:
                    AfType: 1
                    SrcIpv4: 33.0.0.1
                    SrcIpv6:
                    DstIPv4: 255.255.255.255
                    DstIPv6:
                    LinkLocal:
                    Port: 68
                UpData:
                    AccessInterface: GigabitEthernet0/0/0/1
                    CpSubscriberId: 16777218
                    UpSubscriberId: 0
                    UPSubInterfaceId: 0
                    RouterName: automation-userplane
                    AccessVrf: access-vrf-1
                    NASID: NAS-ID-1
                Packet:
                    Payload:
                        BaseLayer:
```

```
                        Operation: 2
                        HardwareType: 1
                        HardwareLen: 6
                        HardwareOpts: 0
                        Xid: 1
                        Secs: 0
                        Flags: 32768
                        ClientIP: 0.0.0.0
                        YourClientIP: 33.0.0.3
                        NextServerIP: 0.0.0.0
                        RelayAgentIP: 0.0.0.0
                        ClientHWAddr: aa:bb:00:00:00:01
                        ServerName:
                        File:
                        Options: {
  Option(MessageType:Offer)
  Option(ClientID:[1 170 187 0 0 0 1])
  Option(SubnetMask:255.255.224.0)
  Option(LeaseTime:90060)
  Option(Timer1:45030)
  Option(Timer2:78802)
  Option(ServerID:33.0.0.1)
}

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

      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.293167
      Message: BNGN4UdpProxyMessage
      Description: Received Packet IPOE, IPC Message from udp-proxy
      Source: bng.udp-proxy.DC.Local.0
      Destination: bng.bng-n4-protocol.DC.Local.0
      PAYLOAD:
        BNGN4UdpProxyMessage:
            BNGN4UdpProxyMessage:
                Type: 6
                L2Data:
                    SrcMac: aabb.0000.0001
                    DstMac: ffff.ffff.ffff
                    Outervlan: 100
                    Innervlan: 200
                    OuterCos: 0
                    InnerCos: 0
                IpAddr:
                    AfType: 1
                    SrcIpv4:
                    SrcIpv6:
                    DstIPv4: 8.8.8.8
                    DstIPv6:
                    LinkLocal:
                    Port: 8000
                UpData:
                    AccessInterface: GigabitEthernet0/0/0/1
                    CpSubscriberId: 0
                    UpSubscriberId: 0
                    UPSubInterfaceId: 0
                    RouterName: automation-userplane
                    AccessVrf: access-vrf-1
                    NASID: NAS-ID-1
                NasInfo:
                    Port: 4
                    Slot: 2
                    Adapter: 5
                    Subslot: 3
```

```
                        Chasis: 1
                        InterfaceType: 1
                    L2TPData:
                        PuntPoliceRate: 0
                        L2TPTos: 0
                        TunnelID: 0
                    Packet:
                        Payload:
                            BaseLayer:
                            Operation: 1
                            HardwareType: 1
                            HardwareLen: 6
                            HardwareOpts: 0
                            Xid: 1
                            Secs: 0
                            Flags: 32768
                            ClientIP: 0.0.0.0
                            YourClientIP: 0.0.0.0
                            NextServerIP: 0.0.0.0
                            RelayAgentIP: 0.0.0.0
                            ClientHWAddr: aa:bb:00:00:00:01
                            ServerName:
                            File:
                            Options: {
  Option(MessageType:Request)
  Option(ClientID:[1 170 187 0 0 0 1])
  Option(ServerID:33.0.0.1)
  Option(RequestIP:33.0.0.3)
}

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

      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.301343
      Message: BNGN4SessionEstablishmentReq
      Description: Sending N4 Session Establishment Request, IPC Message to udp-proxy
      Source: bng.bng-n4-protocol.DC.Local.0
      Destination: bng.udp-proxy.DC.Local.0
      PAYLOAD:
        BNGN4SessionEstablishmentReq:
            BNGN4SessionEstablishmentReq:
                PfcpSessionHeader:
                    Version: 1
                    SeidSet: true
                    MessageType: 50
                    MessageLen: 413
                    SequenceNumber: 5
                    Seid: 0
                    Priority: 1
                NodeID:
                    Valid: true
                    Ip: Afi=v4 Ip=10.105.254.113
                Fseid:
                    Valid: true
                    Seid: 16777218
                    Ipv4: 0.0.0.0
                    Ipv6:
                CreatePdrList:
                    CreatePdrList[0]:
                        Valid: true
                        PdrId:
                            Valid: true
                            RuleId: 1
                        Precedence:
```

```
                    Valid: true
                    Val: 1
                Pdi:
                    Valid: true
                    SrcIface:
                        Valid: true
                        Value: 1
                    TrafficEndptId:
                        Valid: true
                        Val: 1
                FarId:
                    Valid: true
                    Val: 1
                OuterHeaderRemoval:
                    Valid: false
                    Description: 0
                UrrId:
                    Valid: true
                    Val: 1
            CreatePdrList[1]:
                Valid: true
                PdrId:
                    Valid: true
                    RuleId: 2
                Precedence:
                    Valid: true
                    Val: 1
                Pdi:
                    Valid: true
                    SrcIface:
                        Valid: true
                        Value: 2
                    TrafficEndptId:
                        Valid: false
                        Val: 0
                FarId:
                    Valid: true
                    Val: 2
                OuterHeaderRemoval:
                    Valid: false
                    Description: 0
                UrrId:
                    Valid: false
                    Val: 0
        CreateFarList:
            CreateFarList[0]:
                Valid: true
                FarId:
                    Valid: true
                    Val: 1
                ApplyAction:
                    Valid: true
                    Drop: false
                    Forward: true
                    Buffer: false
                    NotifyCP: false
                    Duplicate: false
                ForwParams:
                    Valid: true
                    DestIface:
                        Valid: true
                        Value: 2
                    OuterHeaderCreation:
                        Valid: true
```

```
                CprNSH: false
                TfEndpt: true
                L2tp: false
                Ppp: false
                TunnelID: 0
                SessionID: 0
            DuplParams:
                Valid: false
                DestIface:
                    Valid: false
                    Value: 0
                OuterHeaderCreation:
                    Valid: false
                    Teid: 0
                    Ipv4:
                    Ipv6:
                    PortNum: 0
                IntrInfo:
                    Valid: false
                    InterceptId:
                        Valid: false
                    Dscp:
                        Valid: false
                        Dscp: 0
        CreateFarList[1]:
            Valid: true
            FarId:
                Valid: true
                Val: 2
            ApplyAction:
                Valid: true
                Drop: false
                Forward: true
                Buffer: false
                NotifyCP: false
                Duplicate: false
            ForwParams:
                Valid: true
                DestIface:
                    Valid: true
                    Value: 1
                OuterHeaderCreation:
                    Valid: true
                    CprNSH: false
                    TfEndpt: true
                    L2tp: false
                    Ppp: false
                    TunnelID: 0
                    SessionID: 0
            DuplParams:
                Valid: false
                DestIface:
                    Valid: false
                    Value: 0
                OuterHeaderCreation:
                    Valid: false
                    Teid: 0
                    Ipv4:
                    Ipv6:
                    PortNum: 0
                IntrInfo:
                    Valid: false
                    InterceptId:
                        Valid: false
```

```
                Dscp:
                    Valid: false
                    Dscp: 0
CreateTrafficEndptList:
    CreateTrafficEndptList[0]:
        Valid: true
        Tfid:
            Valid: true
            Val: 1
        AccessPortId:
            Valid: true
            Value: GigabitEthernet0/0/0/1
        UeIPAddr:
            Valid: true
            Flags: 2
            Ipv4Addr: Afi=v4 Ip=33.0.0.3
            Ipv6Addr:
            IPv6PrefixLen: 0
            Ipv6PDAddr:
            Ipv6LLAddr:
        UeMacAddress: aa:bb:00:00:00:01
        PppoeSessId:
            Valid: false
            Value: 0
        AddressFamily:
            Valid: true
            Value: 3
        Cvlan:
            Valid: true
            Pcp: 0
            Dei: 0
            VlanId: 200
        Svaln:
            Valid: true
            Pcp: 0
            Dei: 0
            VlanId: 100
        L2tpTunnel:
            Valid: false
            TunnelEndpoint:
                Valid: false
                Choose: false
                LocalID: 0
                RemoteID: 0
            SessionID:
                Valid: false
                SessionID: 0
                RemoteSessionID: 0
            TunnelFeatures:
                Valid: false
                SetTOS: false
                ReflectTOS: false
                SetDF: false
                ReflectDF: false
                TcpMssAdjust: false
                TunnelStatsEnabled: false
                SessStatsEnabled: false
                TSI: false
                SSI: false
                TosVal: 0
                TcpMssVal: 0
                TunnelStatsInterval: 0
                SessStatsInterval: 0
SubParams:
```

```
        Valid: true
        Stype:
            Valid: true
            Value: 1
        SrgIntfId:
            Valid: false
            Value: 0
        SrgGrpId:
            Valid: false
            Value: 0
        Vrf:
            Valid: true
            Value: automation-vrf
        AccessVrf:
            Valid: false
    CreateURR:
        CreateURR[0]:
            Valid: true
            UrrID:
                Valid: true
                Val: 1
            MeasurementMethod:
                Valid: true
                Event: false
                Volume: true
                Duration: false
            Trigger:
                Valid: true
                PeriodicReporting: true
                VolumeThreshold: false
                TimeThreshold: false
                QuotaHoldingTime: false
                StartOfTraffic: false
                StopOfTraffic: false
                DroppedDlTrafficThreshold: false
                ImmediateReport: false
                VolumeQuota: false
                TimeQuota: false
                LinkedUsageReporting: false
                TerminationReport: true
                MonitoringTime: false
                EnvelopeClosure: false
                MacAddressReporting: false
                EventThreshold: false
                EventQuota: false
                TerminationByUP: false
            MeasurementPeriod:
                Valid: true
                Val: 1940
    Keepalive:
        Valid: false
        Tfid:
            Valid: false
            Val: 0
        Timer:
            Valid: false
            TimeInterval: 0
            RetryCount: 0
        MagicNum:
            Valid: false
            LocalMagicNum: 0
            PeerMagicNum: 0
    CreateQspList:
        CreateQspList[0]:
```

```
                Valid: true
                Service:
                    Valid: true
                    Length: 0
                    Value: automation-feature-template-accounting
                QosIngress:
                    Valid: true
                    Length: 0
                    Name: inpolicy
                    Priority: 0
                QosEgress:
                    Valid: true
                    Length: 0
                    Name: outpolicy
                    Priority: 0
                Stats:
                    Valid: true
                    Value: true
                Spi:
                    Valid: false
                    Value: 0
                PlainQos: false
            CreateACL:
                Valid: false
                Ipv4InACL:
                    Valid: false
                Ipv4OutACL:
                    Valid: false
                Ipv6InACL:
                    Valid: false
                Ipv6OutACL:
                    Valid: false
            CreatePBR:
                Valid: false
                PbrIngress:
                    Valid: false
                    Length: 0
            CreateuRPF:
                Valid: false
                Strictv4: false
                Strictv6: false
                Loosev4: false
                Loosev6: false
            CreateICMP:
                Valid: false
                V4: false
                V6: false
            RemoveICMP:
                Valid: false
                V4: false
                V6: false
            CreateMTU:
                Valid: true
                V4Mtu: 1400
                V6Mtu: 0
                PPPMtu: 0
            TransactionIdentifier:
                Valid: true
                Value: 1
-------------------------------------------------
```

# Configuring Monitor Protocol

Use the following commands to enable protocol monitoring for a subscriber.

**monitor protocol interface** *pcap_interface* **capture-duration** *duration_in_seconds*

**NOTES**:

- **interface** *pcap_interface* : Specifies the packet capture (PCAP) interface. The valid PCAP interfaces are: Packet Forwarding Control Protocol (PFCP), GPRS Tunnelling Protocol User Plane (GTP-U), and Remote Authentication Dial-In User Service (RADIUS).
- **capture-duration** *duration_in_seconds* : Specifies the duration in seconds during which the monitor protocol is enabled. The *duration_in_seconds* can range from 1 to 2147483647 seconds. The default is 300.
- cnBNG uses a custom GTPU packet format. Therefore, packet decode errors are displayed on the screen because the standard decode plugin does not support the cnBNG format. Capture the packet to PCAP and use the cnBNG specific LUA plugin during Wireshark decode.
- Interface names must be entered manually and must match the name mentioned in the description, else the packet capture may fail.
- Only one physical-interface (NIC) packet capture is supported. For PFCP and GTPU this limitation is not applicable as they always run-on a single interface (VIP). However for RADIUS, certain deployments may use different VIPs for Auth/Acct/COA, leading to different physical NICs. Due to the infrastructure limitation, packet-capture can run on only one of the physical-NICs.

### Example

**monitor protocol interface pfcp**

```
InterfaceName = N4:10.86.73.161:8805 |  InterfaceIP = 10.86.73.161 | Filter = (tcp or udp)
 and (port 8805)
<<<<OUTBOUND
from 10.86.73.161:8805 to 10.86.73.162:8805
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2019-10-22 09:22:34.029363 +0000 UTC CaptureLength:72 Length:72
 InterfaceIndex:2 AncillaryData:[]}

Packet Raw Bytes:

0050569c14610050569c8d5c0800450000003a76c5400040111bff0a5649a10a5649a222652265002 6a8262006001a00000004003c0005000a5649a1001300010100600004e159480e

Packet Dump:
 -- FULL PACKET DATA (72 bytes) ------------------------------------
00000000  00 50 56 9c 14 61 00 50  56 9c 8d 5c 08 00 45 00
00000010  00 3a 76 c5 40 00 40 11  1b ff 0a 56 49 a1 0a 56
00000020  49 a2 22 65 22 65 00 26  a8 26 20 06 00 1a 00 00
00000030  00 04 00 3c 00 05 00 0a  56 49 a1 00 13 00 01 01
00000040  00 60 00 04 e1 59 48 0e
--- Layer 1 ---
Ethernet {Contents=[..14..] Payload=[..58..] SrcMAC=00:50:56:9c:8d:5c DstMAC=00:50:56:9c:14:61
 EthernetType=IPv4 Length=0}
00000000  00 50 56 9c 14 61 00 50  56 9c 8d 5c 08 00
--- Layer 2 ---
IPv4 {Contents=[..20..] Payload=[..38..] Version=4 IHL=5 TOS=0 Length=58 Id=30405 Flags=DF
 FragOffset=0 TTL=64 Protocol=UDP Checksum=7167 SrcIP=10.86.73.161 DstIP=10.86.73.162
Options=[] Padding=[]}
00000000  45 00 00 3a 76 c5 40 00  40 11 1b ff 0a 56 49 a1
```

```
00000010  0a 56 49 a2
--- Layer 3 ---
UDP {Contents=[..8..] Payload=[..30..] SrcPort=8805(pfcp) DstPort=8805(pfcp) Length=38
Checksum=43046}
00000000  22 65 22 65 00 26 a8 26                          |"e"e.&.&|
--- Layer 4 ---
Payload 30 byte(s)
00000000  20 06 00 1a 00 00 00 04  00 3c 00 05 00 0a 56 49
00000010  a1 00 13 00 01 01 00 60  00 04 e1 59 48 0e
```

# Copying Log Files

Use the following commands to copy the stored log files externally or on the BNG Ops Center.

These files ether can be copied outside or dumped on the bng-opscenter using the following CLI command.

**monitor subscriber-dump filename <file path got from monitor subscriber-list CLI>**

**Example**:

```
monitor subscriber dump filename
/opt/workspace/logs/monsublogs/none.aabb.0000.0001@automation-userplane_TS_2021-06-09T12:17:33.838574118.txt.sorted
RELEASE_NAMESPACE: 'bng'
Dumping file
'/opt/workspace/logs/monsublogs/none.aabb.0000.0001@automation-userplane_TS_2021-06-09T12:17:33.838574118.txt.sorted'
**** Received 19 messages ******
      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.194843
      Message: BNGN4UdpProxyMessage
      Description: Received Packet IPOE, IPC Message from udp-proxy
      Source: bng.udp-proxy.DC.Local.0
      Destination: bng.bng-n4-protocol.DC.Local.0
      PAYLOAD:
        BNGN4UdpProxyMessage:
            BNGN4UdpProxyMessage:
                Type: 6
                L2Data:
                    SrcMac: aabb.0000.0001
                    DstMac: ffff.ffff.ffff
                    Outervlan: 100
                    Innervlan: 200
                    OuterCos: 0
                    InnerCos: 0
                IpAddr:
                    AfType: 1
                    SrcIpv4:
                    SrcIpv6:
                    DstIPv4: 8.8.8.8
                    DstIPv6:
                    LinkLocal:
                    Port: 8000
                UpData:
                    AccessInterface: GigabitEthernet0/0/0/1
                    CpSubscriberId: 0
                    UpSubscriberId: 0
                    UPSubInterfaceId: 0
                    RouterName: automation-userplane
                    AccessVrf: access-vrf-1
                    NASID: NAS-ID-1
                NasInfo:
                    Port: 4
                    Slot: 2
```

```
                    Adapter: 5
                    Subslot: 3
                    Chasis: 1
                    InterfaceType: 1
                L2TPData:
                    PuntPoliceRate: 0
                    L2TPTos: 0
                    TunnelID: 0
                Packet:
                    Payload:
                        BaseLayer:
                        Operation: 1
                        HardwareType: 1
                        HardwareLen: 6
                        HardwareOpts: 0
                        Xid: 1
                        Secs: 0
                        Flags: 32768
                        ClientIP: 0.0.0.0
                        YourClientIP: 0.0.0.0
                        NextServerIP: 0.0.0.0
                        RelayAgentIP: 0.0.0.0
                        ClientHWAddr: aa:bb:00:00:00:01
                        ServerName:
                        File:
                        Options: {
                                Option(MessageType:Discover)
                                Option(ClientID:[1 170 187 0 0 0 1]).

.


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

      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.205174
      Message: RadiusUdpProxyMsg
      Description: Send Auth/Acct Request Message to UDP-Proxy
      Source: bng.radius-ep.DC.Local.0
      Destination: bng.udp-proxy.DC.Local.0
      PAYLOAD:
        RadiusUdpProxyMsg:
            RadiusUdpProxyMsg:
                SrcIp: 10.105.254.113
                SrcPort: 16384
                DestIp: 10.105.254.114
                DestPort: 1812
                Payload:


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

      Subscriber Id: aa11.0000.0003@asr9k-1
      Timestamp: 2021/06/03 06:26:26.796023
      Message: RadiusUdpProxyMsg
      Description: Send Auth/Acct Request Message to UDP-Proxy
      Source: BNG.radius-ep.DC.Local.0
      Destination: BNG.udp-proxy.DC.Local.0
      PAYLOAD:
        RadiusUdpProxyMsg:
            RadiusUdpProxyMsg:
                SrcIp: 10.1.4.150
                SrcPort: 16384
                DestIp: 10.1.4.151
```

```
                DestPort: 1813
                Payload:
                        Code = AccountingRequest
                        Id = 31
                        Authenticator = [88 13 251 114 225 205 9 68 52 194 48 231 234 226
226 184]
                        User-Name = "cnbng"
                        NAS-IP-Address = 10.1.4.150
                        NAS-Port = 16384
                        Service-Type = 5
                        Framed-IP-Address = 1.0.3.13
                        Nas-Identifier = "CISCO-BNG-ACCT"
                        Acct-Status-Type = 1
                        Acct-Delay-Time = 0
                        Acct-Session-Id = "Local_DC_16777230"
                        Event-Timestamp = 1622701602
                        NAS-Port-Type = 41
                        Acct-Interim-Interval = 300
                        NAS-Port-Id = "asr9k-1/2/3/4/100.200"
                        NAS-IPv6-Address = ::/0
                        Cisco-Vsa_cisco-nas-port = "asr9k-1/2/3/4/100.200"
                        Cisco-Vsa_cisco-dhcp-client-id = 0x01aa1100000003
                        Cisco-Vsa_Cisco AVpair = "client-mac-address=aa11.0000.0003"
                        Cisco-Vsa_Cisco AVpair = "dhcp-class=RJIL_DHCPV4_CLASS_2"
                        Cisco-Vsa_Cisco AVpair = "dhcp-class=RJIL_DHCPv6_CLASS_1"
                        Cisco-Vsa_Cisco AVpair = "accounting-list=aaa-prof1"
                        Cisco-Vsa_Cisco AVpair =
0x646863702d636c69656e742d69643d01aa1100000003
                        Cisco-Vsa_Cisco AVpair = "vrf=ISP"
                PayloadLen: 396
                SubscriberID: aa11.0000.0003@asr9k-1

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

      Subscriber Id: aa11.0000.0003@asr9k-1
      Timestamp: 2021/06/03 06:26:26.800776
      Message: RadiusUdpProxyMsg
      Description: Received Auth/Acct Response Message from UDP-Proxy
      Source: BNG.udp-proxy.DC.Local.0
      Destination: BNG.radius-ep.DC.Local.0
      PAYLOAD:
        RadiusUdpProxyMsg:
            RadiusUdpProxyMsg:
                SrcIp: 10.1.4.151
                SrcPort: 1813
                DestIp: 10.1.4.150
                DestPort: 16384
                Payload:
                        Code = AccountingResponse
                        Id = 31
                        Authenticator = [168 192 147 70 117 31 151 16 237 80 68 105 42 191
 23 186]
                PayloadLen: 20

-------------------------------------------------
 bng#
```

**Note**

- While receiving CoA or DM packets, the RADIUS pod does not have the subscriber-information, instead the information is available only with the BNG-SM pod. Therefore, the packet related session programming N4-SESS-UPDATE TX and RX is dumped on the screen first followed by the CoA or DM TX and RX dump.
- Packet dumps are not captured for PFCP session report request and response.

# Viewing Log Files

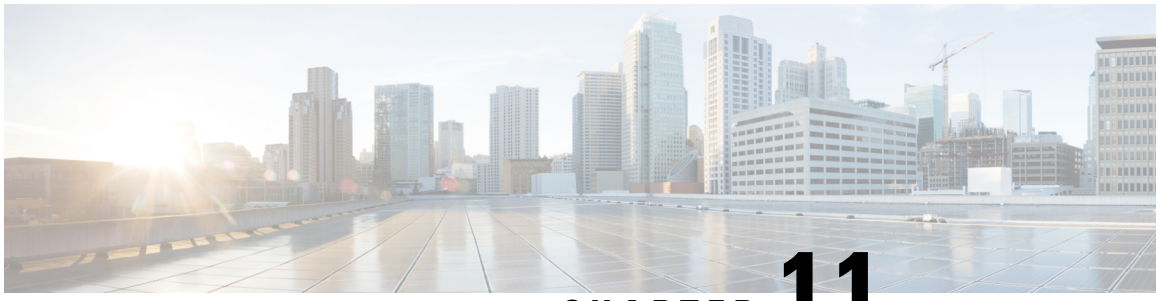
Use the following commands to view the stored log files for a monitor protocol or subscriber.

```
monitor subscriber list
monitor protocol list
```

The following is a sample output for the **monitor subscriber list**.

**Example**:

```
bng# monitor subscriber list
none.aa11.0000.0004*_TS_2021-06-03T06:28:13.564009704.txt.sorted
none.aa11.0000.0003@asr9k-1_TS_2021-06-03T06:26:20.627655233.txt.sorted
none.*_TS_2021-06-03T06:25:04.176857711.txt.sorted
bng#
```



CHAPTER 11

# PPPoE Subscriber Management



# Feature Summary and Revision History

## Summary Data

***Table 31: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | cnBNG |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Disabled - Configuration Required |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

## Revision History

***Table 32: Revision History***

| Revision Details | Release |
|---|---|
| First introduced. | 2021.01.0 |

# Feature Description

Point-to-Point Protocol (PPP) over Ethernet (PPPoE) is a point-to-point link with the subscriber over an Ethernet network where the standard PPP negotiations are used for authentication and IPv4 address assignment. The basic PPPoE is defined in RFC-2516. This RFC defines two distinct stages:

- **Discovery stage:** This sets up a point-to-point session over which PPP can run between two points. For example, between the CPE and Broadband Network Gateway (BNG). This is the PPPoE protocol itself.

  Unlike PPP, the PPPoE discovery protocol defines a client-server relationship with the client initiating the discovery of the server and the subsequent setup of the point-to-point link.

- **Session stage:** This runs over the established point-to-point connection, negotiating the PPP protocols (LCP, Authentication, IPCP) as required for a standard PPP interface.

  The session stage carries the data packets from the PPPoE (this includes PPP protocol negotiation) and the actual data packets to and from the subscriber.

# PPPoE Overview

The cnBNG CP supports the standard PPPoE protocol, as defined in RFC-2516. It implements the PPPoE server functionality, that is, providing PPPoE sessions to subscribers who request them. More specifically, it supports the following functionality:

- Handling incoming PPPoE Active Discovery Initiation (PADI) packets and replying with a PPPoE Active Discovery Offer (PADO) packet when the PADI is valid.
- Handling incoming PPPoE Active Discovery Request (PADR) packets and setting up a PPPoE session for the subscriber when the PADR is valid. It also replies with a PPPoE Active Discovery Session (PADS) with an allocated session-id. When the PADR is not valid (or session setup fails), a PADS is sent containing a zero session-id and an error tag.
- Handling incoming PPPoE Active Discovery Termination (PADT) packets and terminating the corresponding PPPoE sessions.
- Sending a PADT packet to the subscriber when terminating a PPPoE session.

## PPPoE Features

The cnBNG supports the following PPPoE features.

### PPPoE Tag Support

cnBNG supports the following PPPoE tags as defined in RFC-2516.

- Service-Name
- AC-Name tag
- AC-Cookie
- Host-Uniq tag
- Relay-Session-Id tag
- End-Of-List tag
- Vendor-Specific tags
- Error tags
- Max-payload tag

### Interface types

PPPoE is generally supported on all types of Ethernet interfaces. The cloud-native CP supports PPPoE if the configuration is present either on the port identifier, NAS level, or at the router level. The UP is responsible for the interfaces where the PPPoE punt inject towards CP can be enabled.

### CoS Bits

The cnBNG allows configuration of the Class-of-Service (CoS) bits value used in the Ethernet header of PADx packets. This ensure that the PPPoE control packets get treated at a higher priority. The cnBNG CP passes these values in the inject packet and the UP places these CoS values in the PADx packets it forwards towards the CPE.

### Service Selection

The PPPoE Service Selection feature uses service tags to enable a PPPoE server to offer PPPoE clients a selection of different services in the PADO. Then the client chooses one of the services offered and then sends the desired service name in a PADR. This feature enables service providers to offer a variety of services and to charge customers according to the chosen services.

Whenever a PADI is received containing one of the locally configured service-names, the PADO response contains all the configured service-names.

A configuration is also provided to allow the user to disable Service Selection. In this case, the PADO only contains the service-name that was in the original PADI.

### Session Limits

- Mac-limit – max sessions per MAC address.
- Circuit-id-limit – max sessions per circuit Id.
- Outer-vlan-limit – max sessions per outer VLAN.
- Max-limit – total max sessions per UP.

# PPP Overview

The Point-to-Point Protocol provides a standard method for transporting multiprotocol datagrams over point-to-point links. It defines an encapsulation scheme, a link layer control protocol (LCP) and a set of network control protocols (NCPs) for different network protocols that can be transmitted over the PPP link.

The LCP is used to configure and maintain the data link. PPP peers use the LCP to negotiate various link layer properties or characteristics.

An NCP is used to establish and configure the associated network protocol before data packets for the protocol are transmitted. For example, IP Control Protocol (IPCP) is used to negotiate IPv4 addresses between peers.

Between LCP and NCP negotiation phases there is an optional authentication phase that the LCP exchanges are agreed upon. Several different authentication schemes are selected with Challenge Handshake Authentication Protocol (CHAP) being the most prevalent one. The basic PPP protocol is defined in RFC 1661 and there are extensions to it for various features.

## PPP Features

The cnBNG supports the following point-to-point protocols required for bringing up a PPPoE session.

- Link Control Protocol (LCP): This is used for PPP link configuration.
- IP Control Protocol (IPCP): This is used to negotiate IPv4 addresses between peers.
- IPv6 Control Protocol (IPv6CP): This is used to negotiate IPv6 interface ID.
- Password Authentication Protocol (PAP): This is used to verify the identity of the peer by means of a two-way handshake
- Challenge Handshake Authentication Protocol (CHAP): This is used to verify the identity of the peer by means of a three-way handshake.

For more information about the protocols and their negotiation, refer the respective RFCs.

## Address Assignment Strategies

The IPv4 address assignment occurs as part of the IPCP negotiation. The address can be part of the RADIUS profile. Often it is the RADIUS profile that specifies the pool to use and the Control Plane (CP) selects an address from that pool. If neither the address nor pool comes from the RADIUS, the PPP profile configuration (on the box) specifies which pool name to use. This profile is attached to the port identifier where the PPP packets are received.

The IPv6 address assignment occurs in two phases:

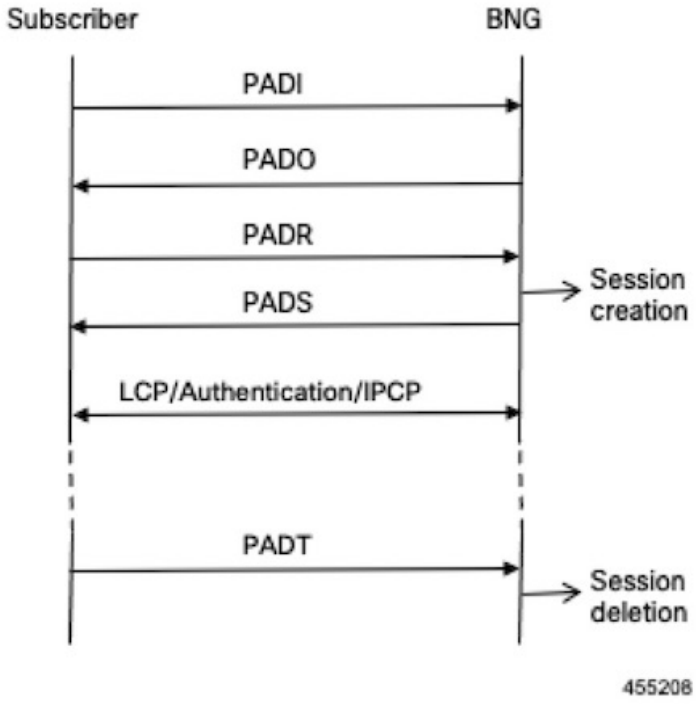
- First, as part of the IPv6CP, the interface-ID is negotiated with the CPE, which is used for link local negotiation.
- Second, after the CPE initiates the DHCPv6 protocol to get IPV6 IANA or IAPD (or both) address allocation, it gets the IPv6 address from either the RADIUS or from a pool.

# How it Works

This section provides a brief of how the PPPoE Subscriber Management feature works.

## PPPoE Handling

The PPPoE discovery-stage protocol consists of basic packet exchange between the subscriber and server (cnBNG). The following illustration displays the flow of events.




In brief, the protocol can be summarized as follows:

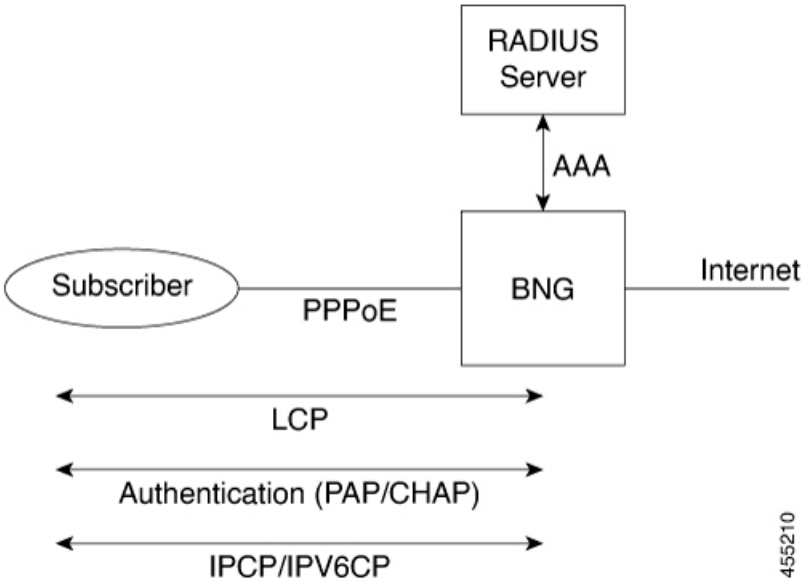
- When the subscriber wishes to establish a PPPoE session, it sends PADI message to the server.
  - The PADI may be multicast, if the subscriber tries to find out if any servers are available.
  - The PADI contains a Service-Name tag, which indicates the service that it wants the server to provide.
- When a server receives a PADI message, it checks if it can provide the service requested to the subscriber. If it can, it replies with a PADO message.
  - The PADO message is unicast to the peer. It contains the Service-Name the client requested.
- When the subscriber receives the PADO messages from the servers, it selects the server to connect to and sends a PADR message.
  - The PADR message is unicast, directed to the specific server with which it wants to establish a session.
  - The PADR message also contains the Service-Name tag.
- When the server receives a PADR message, it checks if it can provide the service to the subscriber.
  - If it can, it chooses a 16-bit Session-Id to identify the session of the subscriber and sets up the necessary state for the subscriber. It then replies with a PADS confirmation, which contains the Session-Id to indicate to the subscriber that the session is established.
  - If it cannot provide a session, it replies with a PADS containing an Error-tag, which indicates the reason it cannot. This PADS contains a zero Session-id.

- After the PADS is sent, the subscriber and server negotiate PPP in the standard way.
- When either the subscriber or the server wants to terminate the session, it sends PADT message to the peer with the Session-Id. This clears up all the states associated with the session.

  This completes the PPPoE discovery stage. the peers can now start the PPP negotiation.

## PPP Handling

The network topology of the PPP is the point-to-point link between the BNG and the subscriber (this link is established during the PPPoE Discovery phase):




The PPPoE subscriber is viewed like any other PPP peer – LCP, Authentication and IPv4CP or IPv6CP (or both) are negotiated to establish the PPP link.

The standard scenario where the BNG terminates both the PPPoE and PPP subscriber session is referred to as PPP Termination and Aggregation (PTA). This distinguishes it from the more complex L2TP Access Concentrator (LAC) and L2TP Network Server (LNS) scenarios where the PPPoE is terminated locally on the BNG but the PPP session is terminated on a separate node from over L2TP to an upstream box known as an LNS.

## Call Flows

This section includes the following high-level call flow.

### PPPoE Bring-Up Call Flow

In cnBNG, the PPPoE and PPP Control Plane runs the overall PTA session bring-up, which includes the PPPoE and PPP negotiation as shown in the following call-flow.

*Figure 10: cnBNG PPPoE Bring-Up Call Flow*

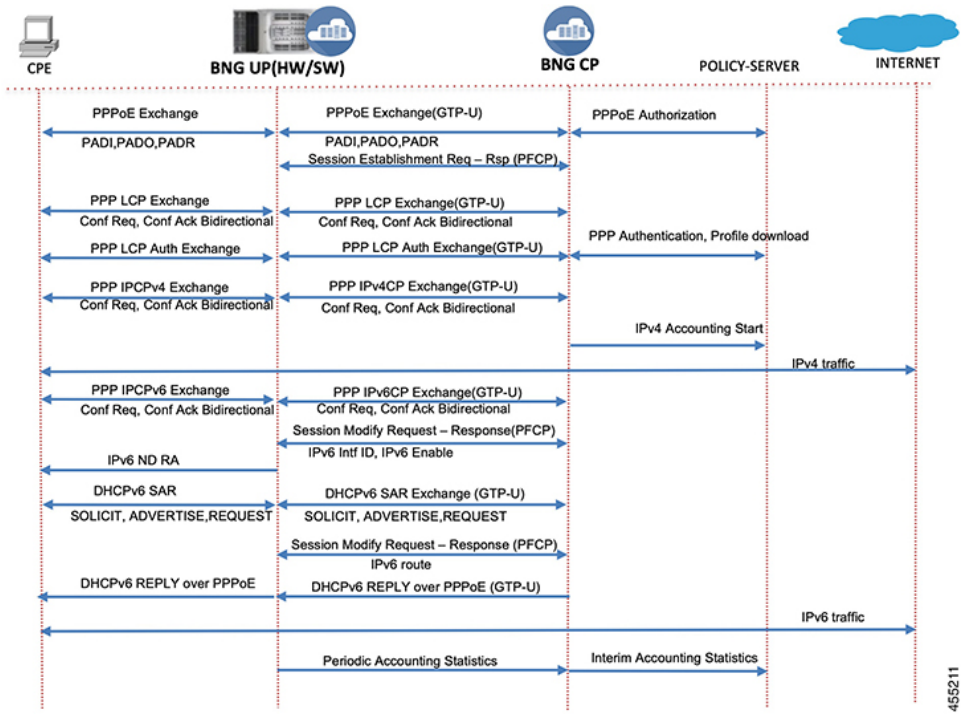



*Table 33: cnBNG PPPoE Call Flow Description*

| Steps | Description |
|---|---|
| 1 | The CPE initiates the PPPoE packet exchange. The BNG-CP verifies the tags, session limits, service name, and so on and creates a PPPoE interface. |
| 2 | The BNG-CP sends a Success PADS message with an assigned PPPoE session ID. |
| 3 | The CPE and BNG-CP negotiate the LCP link parameters and authorization methods to use. |
| 4 | The BNG-CP authenticates the CPE with the provided username and password via AAA and downloads the network level parameters. |
| 5 | The CPE starts the IPv4CP and gets the IPv4 address. The BNG-CP programs the IPv4 route and features on the BNG-UP. Accounting start is initiated for IPv4. |
| 6 | Now bidirectional IPv4 traffic is enabled for the subscriber with the applied features. |
| 7 | Optionally, the CPE starts IPv6CP in case of dual stack. The local and peer interface ID are negotiated and the BNG-UP is programmed to allow link local negotiation to occur. |
| 8 | The BNG-UP completes the link local addressing with the IPv6 ND router advertisement. |

| Steps | Description |
|---|---|
| 9 | The CPE starts the DHCPv6 packet exchange on the negotiated PPPoE session to get the global IPv6 address assignment. |
| 10 | The BNG-CP programs the IPv6 routes and features into the BNG-UP and responds to the CPE with the DHCPv6 Reply packet to acknowledge that the IPv6 is up. At this stage, the session is converted into a dual stack in the CP. |
| 11 | The subscriber can now send and receive IPv6 traffic from the Internet. |
| 12 | The BNG-UP collects and pushes the interim statistics to the BNG-CP. The BNG-CP pushes these statistics to the Policy Plane for billing. |

## Standard Compliance

The PPPoE Subscriber Management feature is aligned with the following standards:

- RFC 1661 Point-to-Point Protocol
- RFC 2516. A Method for Transmitting PPP Over Ethernet (PPPoE)

## Limitations

The PPPoE Subscriber Management feature has the following limitations:

- Only PTA sessions are supported.
- Session throttling is not supported
- Session Limits features is supported only with a single PPPoE instance.
- The PPPoE profile and PPP feature template configuration changes are applied only to the new sessions. These changes are not applied to the existing sessions.
- Update of PPP features via CoA is not supported.

# Configuring the PPPoE Subscriber Management Feature

This section describes how to configure the PPPoE Subscriber Management feature.

Configuring the PPPoE Subscriber Management feature involves the following steps:

1. Creating the PPPoE profile
2. Creating the PPP Feature template

# Creating PPPoE Profile

Use the following commands to create a PPPoE profile and provide the PPPoE protocol specific parameters.

```
config
  profile pppoe pppoe_profile_name
```

```
mtu mtu
service-selection-disable [ true | false ]
max-payload minimum { payload_value } maximum { payload_value }
service-name service_name
ac-name ac_name
ac-cookie ac-cookie_name
session max limit { count } threshold { count }
session mac limit { count } threshold { count }
session circuit-id limit { count } threshold { count }
session outer-vlan limit { count } threshold { count }
timeout-completion period
control-packets priority cos_value
exit
```

**NOTES**:

- **profile pppoe** *pppoe_profile_name*: Specifies the PPPoE profile name.
- **mtu** *mtu*: Specifies the default PPP maximum transmission unit (MTU) value to use if the Max-Payload tag is not provided. The valid values range from 500 to 2000. The default value is 1492.
- **service-selection-disable [ true | false ]**: Enables or disables the advertising of extra service names in the PADO packets. True enables the service and false disables the service. The default value is false.
- **max-payload minimum {** *payload_value* **} maximum {** *payload_value* **}**: Specifies the supported PPPoE service name. Multiple service names can be configured simultaneously. The valid value is an alphanumeric string ranging from 1 to 256. All service names are accepted.
- **service-name** *service_name*: Specifies the supported PPPoE service name. Multiple service names can be configured simultaneously. The valid value is an alphanumeric string ranging from 1 to 256. All service names are accepted.
- **ac-name** *ac_name*: Specifies the access concentrator (AC) to use in the PADO packets. The valid value is an alphanumeric string ranging from 1 to 256. The default ac-name is the router hostname.
- **ac-cookie** *ac-cookie_name*: Specifies the AC-Cookie to use in the PADO packets. The valid value is an alphanumeric string ranging from 1 to 256.
- **session max limit {** *count* **} threshold {** *count* **}**: Specifies the total maximum number of sessions and threshold allowed per User Plane per profile. The valid values range from 1 to 65535. The default value is 65535.
- **session mac limit {** *count* **} threshold {** *count* **}**: Specifies the maximum number of sessions and threshold allowed per UP per peer profile. The valid values range from 1 to 65535. The default value is 65535.

  When the threshold is passed, a syslog is printed as a warning.
- **session circuit-id limit {** *count* **} threshold {** *count* **}**: Specifies the maximum number of sessions and threshold allowed per circuit-id. The valid values range from 1 to 65535. The default value is 65535.

  When the threshold is passed, a syslog is printed as a warning.
- **session outer-vlan limit {** *count* **} threshold {** *count* **}**: Specifies the maximum number of sessions and threshold allowed per UP per peer profile. The valid values range from 1 to 65535. The default value is 65535.

  When the threshold is passed, a syslog is printed as a warning.

- **timeout-completion** *period*: Specifies the maximum time to wait for the session to be completed (an NCP to come up for PTA sessions or the L2TP tunnel to be setup for LAC sessions) before terminating the session. The valid values range from 30 to 600 seconds. The default value is 120 seconds.
- **control-packets priority** *cos_value*: Specifies the CoS to use in the PADx packets. The valid values range from 0 to 7. The default CoS bits are used.

# Creating the PPP Feature Template

Use the following commands to create a PPP feature template.

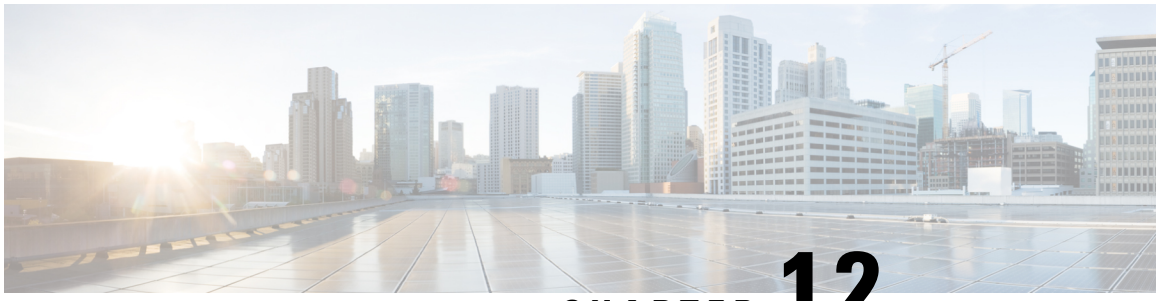
**Note** The PPP feature template allows per subscriber PPP parameters.

```
config
  profile feature-template feature_template_name
  ppp
    authentication { chap  | pap }
    chap hostname chap_hostname
    chap password chap_password
    ipcp dns ipv4_address
    ipcp peer-address-pool ipam_pool_name
    ipcp renegotiation ignore
    ipcp wins ipv4_address
    ipcpv6 renegotiation ignore
    ipcp wins ipv4_address
    max-bad-auth count
    max-configure count
    max-failure count
    pap accept-null-password
    timeout absolute seconds
    timeout authentication seconds
    timeout retry seconds
    keepalive interval secondsretryseconds[ disable ]
    exit
```

**NOTES**:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **ppp**: Enters the PPP Configuration mode to configure the PPP feature.
- **authentication { chap | pap }**: Specifies the authentication type as CHAP or PAP.
- **chap hostname** *chap_hostname*: Specifies the hostname to use for CHAP authentication. The valid values range from 1 to 64. The default value is the router hostname.
- **chap password** *chap_password*: Specifies the password to use for CHAP authentication.
- **ipcp dns** *ipv4_address*: Specifies the DNS address to use for the peer.
- **ipcp peer-address-pool** *ipam_pool_name*: Specifies the address pool to use to obtain an IPv4 address for the peer.

- **ipcp renegotiation ignore**: Specifies to ignore the attempts of the peer to renegotiate IPCP. The entire PPPoE session is terminated on renegotiation.
- **ipcp wins** *ipv4_address*: Specifies the Windows Internet Name Service (WINS) address to use for the peer.
- **max-bad-auth** *count*: Specifies the maximum authentication failures to allow. The valid values range from 0 to 10. The default value is 0.
- **max-configure** *count*: Specifies the maximum number of Conf-Reqs to send without a response. The valid values range from 4 to 20. The default value is 10.
- **max-failure** *count*: Specifies the maximum number of Conf-Naks to send. The valid values range from 2 to 10. The default value is 5.
- **pap accept-null-password**:Accepts the null password feature for PAP.
- **max-failure** *count*: Specifies the maximum number of Conf-Naks to send. The valid values range from 2 to 10. The default value is 5.
- **timeout absolute** *seconds*: Specifies the absolute timeout for a PPP session. The valid values range from 0 to 70000000 minutes.
- **timeout authentication** *seconds*: Specifies the total time to allow for authentication to complete. The valid values range from 3 to 30 seconds. The default value is 10.
- **timeout retry** *seconds*: Specifies the maximum time to wait for a response to a Conf-Req. The valid values range from 1 to 10 seconds. The default value is 3.
- **keepalive interval** *seconds* **retry** *seconds* **[ disable ]**: Specifies the keepalive interval and the retry attempts for the subscribers. The valid values range from 10 to 120 seconds for the keepalive interval. The default is 60 seconds. The valid values range from 1 to 255 for the retry attempt. The default value is 5 counts.



CHAPTER 12

# Subscriber Manager



# Feature Summary and Revision History

## Summary Data

***Table 34: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | cnBNG |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Disabled - Configuration Required |
| Related Changes in this Release | Not Applicable |
| Related Documentation | *Cloud Native BNG Control Plane Command Reference Guide* |

## Revision History

***Table 35: Revision History***

| Revision Details | Release |
|---|---|
| First introduced. | 2021.01.0 |

# Feature Description

In the Subscriber Manager (SM) context, a subscriber is a binding between the cnBNG Control Plane (CP) and a single subscriber end device. The SM is designed to provide a generic mechanism to connect edge subscribers to services enabling features. Subscribers are identified, authenticated, authorized, and accounted for in the SM.

**Note** The Subscriber Manager is also referred to as the Session Manager.

The following is a high-level list of the SM functionalities:

- Provides a generic mechanism for different Broadband Access Protocols such as DHCP and PPPoE.
- Provides an interface with off-box Radius servers using policy-plane to meet protocol and network provisioning requirements.
- Supports different subscriber lifecycle events such as CoA, idle timeout processing, and periodic reauthorization.
- Provides support for configuring subscriber lifecycle events that help customer define the subscriber behavior for the different subscriber lifecycle events.
- Derives per subscriber configuration from multiple sources.
- Maintains the subscriber state and subscriber configuration in a centralized session database.
- Interacts with the User Plane (UP) for subscriber session creation and subscriber feature configurations.

Subscriber features that are configured on cnBNG enable service providers to deploy certain specific functionalities like restricting the use of certain network resources, allowing Law Enforcement Agencies (LEAs) to conduct electronic surveillance, and so on.

### Subscriber Features

The cnBNG supports the following subscriber features on the UP. For details, see the latest version of the Broadband Network Gateway Configuration Guide for Cisco ASR 9000 Series Routers listed here: https://www.cisco.com/c/en/us/support/routers/asr-9000-series-aggregation-services-routers/products-installation-and-configuration-guides-list.html.

- IPv4 or IPv6
  - Maximum Transmission Unit (MTU)
  - Unicast Reverse Path Forwarding (URPF)
  - Internet Control Message Protocol (ICMP)
- Access Control List (ACL)
  - Input ACL (IPv4 or IPv6)
  - Output ACL (IPv4 or IPv6)

- QoS (Quality of Service)
  - Input (policing)
  - Output (policing, shaping)
  - Policy merging (up to 6 policy maps and 10 class maps, including the default)
- Policy-based Routing (PBR)
  - Input policy (HTTP redirect)
- Accounting
  - Session Accounting
    - Periodic accounting
  - Service Accounting
    - Periodic accounting

To configure subscriber features, see Configuring Subscriber Manager Features, on page 127.

## How it Works

This section provides a brief about how the Subscriber Manager works.

The SM functionality is hosted in a SM pod having one container in it. The SM pod communicates with the BNG Ops Center, policy-plane, and PFCP-EP pods using the APP infrastructure inter-pod communication (IPC).

The Subscriber Microservices Infrastructure (SMI) instantiates the SM pod. There can be more than one SM pod in the cluster. Each SM pod instance is independent. The per subscriber data is stored in a centralized database such that any SM pod can access this data.

# Configuring Subscriber Manager Features

This section describes how to configure Subscriber Manager features on the CP.

The configuration of the Subscriber Manager features involves the following procedures:

- Configuring the HTTPR Policy Name, on page 128
- Configuring IPv4 Options, on page 128
- Configuring IPv6 Options, on page 129
- Configuring QoS Parameters
- Configuring the VRF Name, on page 130
- Configuring a Subscriber Profile, on page 130

**Note**

- To configure PPP feature options, see Creating the PPP Feature Template, on page 122
- To configure service accounting, see Configuring Service Accounting, on page 134
- To configure session accounting, see Configuring Session Accounting, on page 134

# Configuring the HTTPR Policy Name

Use the following commands to configure the Policy Based Routing (PBR) HTTP Redirect (HTTPR) policy name.

```
config
  profile feature-template feature_template_name
  httpr-policy httpr_policy_name
  exit
```

**NOTES**:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **httpr-policy** *httpr_policy_name*: Specifies the PBR HTTPR policy name. The *httpr_policy_name* value can range from 1 to 128 characters.

# Configuring IPv4 Options

Use the following commands to configure IPv4 options.

```
config
  profile feature-template feature_template_name
  ipv4
    disable-unreachables
    egress-acl string
    ingress-acl string
    mtu mtu_bytes
    verify-unicast-source reachable-via-rx
    exit
```

**NOTES**:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **ipv4**: Enters the IPv4 Configuration mode to configure the IPv4 features.
- **disable-unreachables**: Disables sending the Internet Control Message Protocol (ICMP) Unreachable messages.
- **egress-acl** *string*: Specifies the IPv4-based egress Access Control List (ACL) list. The supported length of the *string* ranges from 1 to 128 characters.
- **ingress-acl** *string*: Specifies the IPv4-based ingress ACL list. The supported length of the *string* ranges from 1 to 128 characters.

- **mtu** *mtu_bytes*: Specifies the maximum transmission unit (MTU). The supported *mtu_bytes* value can range from 68 to 65535 bytes.
- **verify-unicast-source reachable-via-rx**: Enables per packet validation for unicast. The source is reachable via the interface on which packet is received.

# Configuring IPv6 Options

Use the following commands to configure IPv6 options.

```
config
  profile feature-template feature_template_name
  ipv6
    disable-unreachables
    egress-acl string
    ingress-acl string
    mtu mtu_bytes
    verify-unicast-source reachable-via-rx
    exit
```

**NOTES**:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **ipv6**: Enters the IPv6 Configuration mode to configure the IPv6 features.
- **disable-unreachables**: Disables sending the Internet Control Message Protocol (ICMP) Unreachable messages.
- **egress-acl** *string*: Specifies the IPv6-based egress Access Control List (ACL) list. The supported length of the *string* ranges from 1 to 128 characters.
- **ingress-acl** *string*: Specifies the IPv6-based ingress ACL list. The supported length of the *string* ranges from 1 to 128 characters.
- **mtu** *mtu_bytes*: Specifies the maximum transmission unit (MTU). The supported *mtu_bytes* value can range from 68 to 65535 bytes.
- **verify-unicast-source reachable-via-rx**: Enables per packet validation for unicast. The source is reachable via the interface on which packet is received.

# Configuring QoS Parameters

Use the following commands to configure the Quality of Service (QoS) parameters.

```
config
  profile feature-template feature_template_name
  qos
    in-policy qos_input_policy_name
    merge-level integer
    out-policy qos_output_policy_name
    exit
```

**NOTES**:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **qos**: Enters the QoS Configuration mode to configure the parameters.
- **in-policy** *qos_input_policy_name*: Specifies the QoS input policy name. The supported length of the *qos_input_policy_name* ranges from 1 to 128 characters.
- **merge-level** *integer*: Enables or disables the merge level. A merge value of 0 disables the merge-level. Any value greater than 0, enables the merge level.
- **out-policy** *qos_output_policy_name*: Specifies the QoS output policy name. The supported length of the *qos_output_policy_name* ranges from 1 to 128 characters.

# Configuring the VRF Name

Use the following commands to configure the virtual routing and forwarding (VRF) name.

```
config
  profile feature-template feature_template_name
  vrf-name vrf_name
  exit
```

**NOTES**:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **vrf-name** *vrf_name*: Specifies the VRF name. The supported length of the *vrf_name* ranges from 1 to 128 characters.

# Configuring a Subscriber Profile

Use the following commands to create a subscriber profile.

```
config
  profile subscriber subscriber_profile
    aaa { authenticate aaa_profile_for_authentication
        | authorize aaa_profile_for_authorization }
    activate-feature-template feature_template_name
    apply-all-class
    class class_name
      aaa aaa_profile_for_authentication | authorize aaa_profile_for_authorization
      activate-feature-template feature_template_name
      matches
        match { protocol { dhcp | ppp } } | username
        { ascii ascii_string |
        regex reg-exp string } |
        source-mac { ascii ascii_string |
        regex reg-exp string } |
        circuit-id { ascii  ascii_string |
        regex reg-exp string } |
        remote-id { ascii ascii_string |
        regex reg-exp string }
        match-type { all match { protocol | username | source-mac |
```

```
            circuit-id | remote-id } | any match { protocol | username |
            source-mac | circuit-id | remote-id } }
            exit
         dhcp-profile dhcp_profile_name
         event event_name
         pppoe-profile pppoe_profile_name
         session-type { ipv4 | ipv4v6 | ipv6 }
         exit

configure
  profile subscriber subscriber_profile
     aaa { authenticate aaa_profile_for_authentication |
         authorize aaa_profile_for_authorization }
     activate-feature-template feature_template_name
     apply-all-class
     class class_name
        aaa aaa_profile_for_authentication | authorize aaa_profile_for_authorization
        activate-feature-template  feature_template_name
        matches
          match { protocol { dhcp | ppp } } | username { ascii
                ascii_string  | regex reg-exp string}
                | source-mac { ascii ascii_string
                | regex reg-exp string } |
                circuit-id { ascii  ascii_string
                | regex reg-exp string } |
                remote-id { ascii ascii_string
                | regex reg-exp string }
                match-type { all match { protocol | username |
                source-mac | circuit-id | remote-id } | any match {
                protocol | username | source-mac | circuit-id
                | remote-id } }
          exit
        dhcp-profile  dhcp_profile_name
        event session-activate { aaa { authenticate | authorize } |
                               activate-feature-templates
                               feature_templates_list
                               | apply-all-class | class class_name
                               | deactivate-feature-templates
                               feature_templates_list
        pppoe-profile pppoe_profile_name
        session-type { ipv4 | ipv4v6 | ipv6 }
        exit
```

**NOTES**:

- **profile subscriber** *subscriber_profile_name*: Specifies the profile subscriber name and enters the Profile Subscriber Configuraton mode.
- **aaa { authenticate** *aaa_profile_for_authentication* **| authorize** *aaa_profile_for_authorization* **}**: Specifies the AAA profile to associate for authentication and authorization.
- **activate-feature-templates** *feature_template_name*: Specifies the list of feature-templates in sequence for activation.

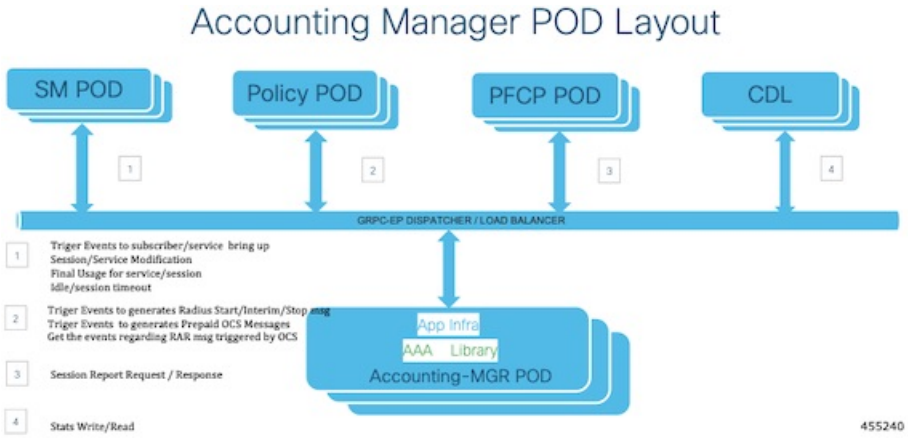
- **apply-all-class**: Applies all classes that are enabled.
- **class** *class_name* : Specifies the subscriber class name.
- **matches**: Enters the matches Configuration sub-mode to specify the match values.
  - **match { protocol { dhcp | ppp } | username { ascii** *ascii_string* **| regex** *reg-exp string* **} | source-mac { ascii** *ascii_string* **| regex** *reg-exp string* **} | { circuit-id { ascii** *ascii_string* **|regex** *reg-exp string* **} | remote-id { ascii** *ascii_string* **| regex** *reg-exp string* **} }**: Specifies the list of match values.
    - **match { protocol { dhcp | ppp }**: Specifies the match protocol as DHCP or PPP.
    - **username { ascii** *ascii_string* **| regex** *reg-exp string* **}**: Specifies the username in ascii format or regular express (reg-exp) string.
    - **source-mac { ascii** *ascii_string* **| regex** *reg-exp string* **}**: Specifies the source MAC address in ascii format or regular express (reg-exp) string.
    - **remote-id { ascii** *ascii_string* **| regex** *reg-exp string* **}**: Specifies the remote identifier in ascii format or regular express (reg-exp) string.
    - **circuit-id { ascii** *ascii_string* **| regex** *reg-exp string* **}**: Specifies the circuit identifier in ascii format or regular express (reg-exp) string.
    - **match-type { all match { protocol | username | source-mac | circuit-id | remote-id } | any match { protocol | username | source-mac | circuit-id | remote-id } }**: Specifies the match key and value for matching any or all of the options: protocol, username, source-mac, circuit-id, and remote-id.
- **dhcp-profile** *dhcp_profile_name*: Associates the DHCP first sign of life (FSOL) profile.
- **event** *event_list_name*: Specifies the event name.
- **pppoe-profile** *pppoe_profile_name*: Associates the PPPoE FSOL profile.
- **session-type { ipv4 | ipv4v6 | ipv6 }**: Specifies the allowed session-types as IPv4, IPv4v6, and IPv6.

# Subscriber Accounting Functions

## Feature Description

The Accounting Manager handles the Subscriber Accounting functions in the cnBNG CP. The Accounting function includes features that track traffic either in volume or duration. It provides accounting information for subscribers on a session or per service. The Accounting function determines the length and duration of a given service that a subscriber has used. Certain regulations require service providers to account for services they provide to the subscriber.

The following figure illustrates the Accounting Manager external interfaces.




The Accounting Manager in cnBNG supports the following forms of accounting:

### Service Accounting

ISPs can offer different tiered services to their subscribers with the ability to move between different tiers. Different tiers could correspond to different bandwidths offered to the subscriber. A subscriber can enable a new service that corresponds to temporarily moving from one tier of service to another. ISPs need to keep track of when a new service is enabled and how long it is active for each subscriber. Often there might be a need to count the number of packets and bytes associated with a service. Both of these forms of accountng are referred to as service accounting. When service accounting is enabled, BNG sends a Service-Start request when service is activated and a Service-Stop request when the service is deactivated. A timestamp is sent with both the actions. Service-Stop can also contain statistics associated with the service.

To configure Service Accounting, see Configuring Service Accounting, on page 134.

### Session Accounting

When Session Accounting is activated, an Accounting-Start request is sent to AAA when the session is started. When the session is terminated, an Accounting-Stop request is sent. The Accounting-Stop request contains the final session accounting statistics (packets, bytes in, bytes out). An “interim” session accounting can be optionally activated that sends Interim-Updates periodically while the session is active. These updates provide the current session statistics accumulated since the start of the session.

Session Accounting is configured directly on the template.

To configure Session Accounting, see Configuring Session Accounting, on page 134.

## Limitations and Restrictions

The Subscriber Accounting Function has the following limitation in this release:

- An interim Interval of zero is not supported.
- AAA profile change at service level is not supported.
- Service-level attributes changes are not supported after service bring-up.
- Session accounting is mandatory to enable Service accounting due to User Plane (UP) (asr9k) limitation.
- Session and Service Accounting enable or disable is not supported after session or service is up because of UP limitations. Session Accounting must be enabled only during session bring-up.

# Configuring Subscriber Accounting Functions

This section describes how to configure the Subscriber Accounting Functions.

The configuration of the Subscriber Accounting Functions involve the following procedures:

- Configuring Service Accounting
- Configuring Session Accounting

## Configuring Service Accounting

Use the following commands to configure service accounting.

```
config
  profile feature-template feature-template
  service accounting
    aaa-profile aaa_profile_name
    enable
    periodic-interval interval_in_seconds
    exit
```

**NOTES**:

- **profile feature-template** *feature-template*: Specifies the profile feature template name and enters Feature-Template Configuration mode.
- **service accounting**: Enters the Service Configuration mode to configure service accounting for a AAA profile.
- **aaa-profile** *aaa_profile_name*: Specifies the AAA profile to use for service accounting.
- **enable**: Enables service accounting for the specified AAA profile.
- **periodic-interval** *interval_in_seconds*: Specifies the interim interval in seconds. The valid values range from 60 to 4320000 seconds.

## Configuring Session Accounting

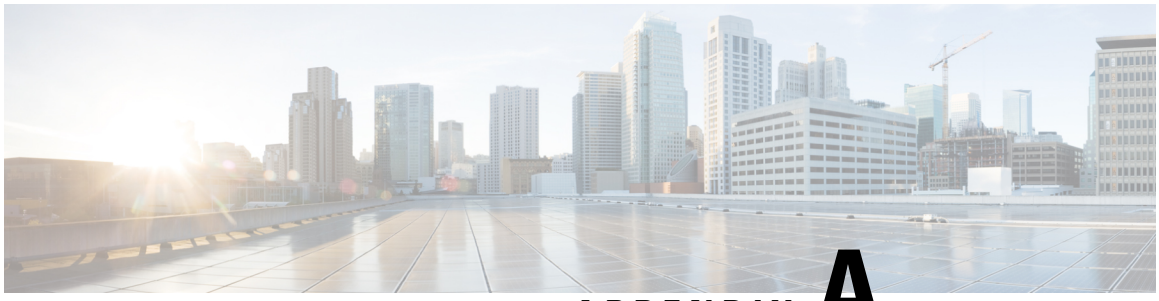
Use the following commands to configure session accounting.

```
config
  profile feature-template feature-template
  session accounting
    aaa-profile aaa_profile_name
    dual-stack-delay delay_in_seconds
    enable
    periodic-interval interval_in_seconds
    exit
```

**NOTES**:

- **profile feature-template** *feature-template*: Specifies the profile feature template name and enters Feature-Template Configuration mode.

- **session accounting**: Enters the Session Configuration mode to configure session accounting for a AAA profile.
- **aaa-profile** *aaa_profile_name*: Specifies the AAA profile to use for session accounting.
- **dual-stack-delay** *delay_in_seconds*: Specifies the dual stack set delay time in seconds. The valid values range from 1 to 30 seconds.
- **enable**: Enables session accounting for the specified AAA profile.
- **periodic-interval** *interval_in_seconds*: Specifies the interim interval in seconds. The valid values range from 60 to 4320000 seconds.



APPENDIX A

# RADIUS Attributes

Remote Authentication Dial-In User Service (RADIUS) attributes are used to define specific authentication, authorization, and accounting (AAA) elements in a user profile, which is stored on the RADIUS daemon.

This appendix describes the following types of RADIUS attributes supported in Broadband Network Gateway (BNG):



# RADIUS IETF Attributes

### IETF Attributes Versus VSAs

RADIUS Internet Engineering Task Force (IETF) attributes are the original set of 255 standard attributes that are used to communicate AAA information between a client and a server. Because IETF attributes are standard, the attribute data is predefined and well known; thus all clients and servers who exchange AAA information via IETF attributes must agree on attribute data such as the exact meaning of the attributes and the general bounds of the values for each attribute.

RADIUS vendor-specific attributes (VSAs) derived from one IETF attribute-vendor-specific (attribute 26). Attribute 26 allows a vendor to create an additional 255 attributes however they wish. That is, a vendor can create an attribute that does not match the data of any IETF attribute and encapsulate it behind attribute 26; thus, the newly created attribute is accepted if the user accepts attribute 26.

***Table 36: Supported RADIUS IETF Attributes***

| Name | Value | Type |
|---|---|---|
| Acct-Delay-Time | integer | 41 |
| Acct-Input-Giga-Words | integer | 52 |
| Acct-Input-Octets | integer | 42 |
| Acct-Input-Packets | integer | 47 |

| Name | Value | Type |
|---|---|---|
| Acct-Interim-Interval | integer | 85 |
| Acct-Link-Count | integer | 51 |
| Acct-Output-Giga-Words | integer | 53 |
| Acct-Output-Octets | integer | 43 |
| Acct-Output-Packets | integer | 48 |
| Acct-Status-Type | integer | 40 |
| Acct-Terminate-Cause | integer | 49 |
| CHAP-Challenge | binary | 40 |
| CHAP-Password | binary | 3 |
| Delegated-IPv6-Prefix | binary | 123 |
| Dynamic-Author-Error-Cause | integer | 101 |
| Event-Timestamp | integer | 55 |
| Framed-Interface-Id | binary | 96 |
| Framed-IP-Address | ipv4addr | 8 |
| Framed-IPv6-Route | string | 99 |
| Framed-Pool | string | 88 |
| Framed-Protocol | integer | 7 |
| Framed-Route | string | 22 |
| Nas-Identifier | string | 32 |
| NAS-IP-Address | ipv4addr | 4 |
| NAS-IPv6-Address | string | 95 |
| NAS-Port | integer | 5 |
| Reply-Message | binary | 18 |
| Service-Type | integer | 6 |
| Stateful-IPv6-Address-Pool | binary | 123 |
| X-Ascend-Client-Primary-DNS | ipv4addr | 135 |
| X-Ascend-Client-Secondary-DNS | ipv4addr | 136 |

# RADIUS Vendor-Specific Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the

vendor-specific attribute (attribute 26). Attribute 26 encapsulates vendor specific attributes, thereby, allowing vendors to support their own extended attributes otherwise not suitable for general use.

The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of this format:

```
protocol : attribute sep value *
```

"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization; protocols that can be used include IP, IPX, VPDN, VOIP, SHELL, RSVP, SIP, AIRNET, OUTBOUND. "Attribute" and "value" are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS.

The following example shows how to cause a user logging in from a network access server to have immediate access to EXEC commands:

```
cisco-avpair= "shell:priv-lvl=15"
```

The following example shows how to configure avpair aaa attribute to enable IPv6 router advertisements from an IPv4 subscriber interface:

```
Cisco-avpair= "ipv6:start-ra-on-ipv6-enable=1"
```

Attribute 26 contains these three elements:

- Type
- Length
- String (also known as data)
  - Vendor-ID
  - Vendor-Type
  - Vendor-Length
  - Vendor-Data

**Note** It is up to the vendor to specify the format of their VSA. The Attribute-Specific field (also known as Vendor-Data) is dependent on the vendor's definition of that attribute.

***Table 37: Supported Cisco Vendor-Specific RADIUS Attributes***

| Name | Value | Type | Present in AAA message type |
|---|---|---|---|
| accounting-list | string | 1 | Access-accept, CoA, Accounting-request |
| acct-input-gigawords-ipv4 | integer | 1 | Accounting-request |
| acct-input-octets-ipv4 | integer | 1 | Accounting-request |
| acct-input-packets-ipv4 | integer | 1 | Accounting-request |

| Name | Value | Type | Present in AAA message type |
|---|---|---|---|
| acct-input-gigawords-ipv6 | integer | 1 | Accounting-request |
| acct-input-octets-ipv6 | integer | 1 | Accounting-request |
| acct-input-packets-ipv6 | integer | 1 | Accounting-request |
| acct-output-gigawords-ipv4 | integer | 1 | Accounting-request |
| acct-output-octets-ipv4 | integer | 1 | Accounting-request |
| acct-output-packets-ipv4 | integer | 1 | Accounting-request |
| acct-output-gigawords-ipv6 | integer | 1 | Accounting-request |
| acct-output-octets-ipv6 | integer | 1 | Accounting-request |
| acct-output-packets-ipv6 | integer | 1 | Accounting-request |
| addrv6 | string | 1 | Access-accept, Accounting-request |
| circuit-id-tag | string | 1 | Access-accept, Accounting-request |
| cisco-dhcp-subscriber-id | string | 65 | Access-request |
| cisco-dhcp-user-class | string | 47 | Access-request |
| cisco-dhcp-vendor-class | string | 48 | Access-request |
| cisco-nas-port | string | 2 | Access-accept, Accounting-request |
| cisco-vsa-in-acl | string | 57 | Access-accept, CoA |
| cisco-vsa-ipv6-in-acl | string | 61 | Access-accept, CoA |
| cisco-vsa-ipv6-out-acl | string | 62 | Access-accept, CoA |
| cisco-vsa-out-acl | string | 58 | Access-accept, CoA |
| cisco-vsa-service-name | string | 51 | Access-accept |
| cisco-vsa-sub-activate-service | string | 60 | Access-accept, CoA |
| cisco-vsa-sub-deactivate-service | string | 63 | Access-accept, CoA |
| cisco-vsa-sub-pbr-policy-in | string | 59 | Access-accept, CoA |
| client-mac-address | string | 1 | Access-accept, Accounting-request |
| command | string | 1 | CoA |
| connect-progress | string | 1 | Accounting-request |
| delegated-ipv6-pool | string | 1 | Access-accept |
| dhcp-class | string | 1 | Access-accept |
| dhcp-client-id | string | 1 | Accounting-request |

| Name | Value | Type | Present in AAA message type |
|---|---|---|---|
| dhcp-vendor-class | string | 1 | Access-request, Accounting-request |
| disc-cause-ext | string | 1 | Accounting-request |
| disconnect-cause | string | 1 | Accounting-request |
| dual-stack-delay | integer | 1 | Access-accept |
| inacl | string | 1 | Access-accept |
| intercept-id | integer | 1 | Access-accept |
| ip-addresses | string | 1 | Access-request, Accounting-request |
| ipv6_inacl | string | 1 | Access-accept, CoA |
| ipv6_outacl | string | 1 | Access-accept, CoA |
| ipv6-dns-servers-addr | string | 1 | Access-accept |
| ipv6-mtu | integer | 1 | Access-accept |
| ipv6-strict-rpf | integer | 1 | Access-accept |
| ipv6-unreachable | integer | 1 | Access-accept |
| md-dscp | integer | 1 | Access-accept |
| md-ip-addr | ipaddr | 1 | Access-accept |
| md-port | integer | 1 | Access-accept |
| outacl | string | 1 | Access-accept |
| parent-session-id | string | 1 | Accounting-request |
| pppoe_session_id | integer | 1 | Accounting-request |
| primary-dns | ipaddr | 1 | Access-accept |
| remote-id-tag | string | 1 | Access-request, Accounting-request |
| sa | string | 1 | Access-accept, CoA |
| sd | string | 1 | RADIUS CoA |
| secondary-dns | ipaddr | 1 | Access-accept |
| service-name | string | 1 | Accounting-request |
| Stateful-IPv6-Address-Pool | string | 1 | Access-accept |
| sub-pbr-policy-in | string | 1 | Access-accept, CoA |
| username | string | 1 | Access-request, Accounting-request |
| vrf | string | 1 | Access-accept |

# Vendor-Specific Attributes for Account Operations

*Table 38: Supported Vendor-Specific Attributes for Account Operations*

| RADIUS AVP | Value | Type | Action |
|---|---|---|---|
| subscriber:command=account-update | string | 1 | account update |
| subscriber:sa=<service-name> | string | 1 | service activate |
| subscriber:sd=<service-name> | string | 1 | service de-activate |

# RADIUS ADSL Attributes

*Table 39: Supported RADIUS ADSL Attributes*

| Name | Value | Type |
|---|---|---|
| Agent-Circuit-Id | string | 1 |
| Agent-Remote-Id | string | 2 |

# RADIUS ASCEND Attributes

*Table 40: Supported RADIUS Ascend Attributes*

| Name | Value | Type |
|---|---|---|
| Ascend-Client-Primary-DNS | ipv4addr | 135 |
| Ascend-Client-Secondary-DNS | ipv4addr | 136 |
| Ascend-Connection-Progress | integer | 196 |
| Ascend-Disconnect-Cause | integer | 195 |

# RADIUS Disconnect-Cause Attributes

Disconnect-cause attribute values specify the reason a connection was taken offline. The attribute values are sent in Accounting request packets. These values are sent at the end of a session, even if the session fails to be authenticated. If the session is not authenticated, the attribute can cause stop records to be generated without first generating start records.

lists the cause codes, values, and descriptions for the Disconnect-Cause (195) attribute.

**Note** The Disconnect-Cause is incremented by 1000 when it is used in RADIUS AVPairs; for example, disc-cause 4 becomes 1004.

*Table 41: Supported Disconnect-Cause Attributes*

| Cause Code | Value | Description |
|---|---|---|
| 2 | Unknown | Reason unknown. |
| 3 | Call-Disconnect | The call has been disconnected. |
| 11 | Lost-Carrier | Loss of carrier. |
| 21 | Idle-Timeout | Timeout waiting for user input.<br>**Note** Codes 21, 100, 101, 102, and 120 apply to all session types. |
| 28 | EXEC-Process-Destroyed | EXEC process destroyed. |
| 33 | Insufficient-Resources | Insufficient resources. |
| 40 | Timeout-PPP-LCP | PPP LCP negotiation timed out.<br>**Note** Codes 40 through 49 apply to PPP sessions. |
| 41 | Failed-PPP-LCP-Negotiation | PPP LCP negotiation failed. |
| 42 | Failed-PPP-PAP-Auth-Fail | PPP PAP authentication failed. |
| 45 | PPP-Remote-Terminate | PPP received a Terminate Request from remote end. |
| 47 | NCP-Closed-PPP | PPP session closed because there were no NCPs open. |
| 52 | Invalid-IP-Address | IP address is not valid for Telnet host. |
| 100 | Session-Timeout | Session timed out. |
| 150 | RADIUS-Disconnect | Disconnected by RADIUS request. |
| 151 | Local-Admin-Disconnect | Administrative disconnect. |
| 170 | PPP-Authentication-Timeout | PPP authentication timed out. |

RADIUS Attributes