Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide, Release 6.0.x

First Published: 2016-04-28

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

Preface

From Release 6.1.2 onwards, Cisco introduces support for the 64-bit Linux-based IOS XR operating system. Extensive feature parity is maintained between the 32-bit and 64-bit environments. Unless explicitly marked otherwise, the contents of this document are applicable for both the environments. For more details on Cisco IOS XR 64 bit, refer to the Release Notes for Cisco ASR 9000 Series Routers, Release 6.1.2 document.

This Preface contains these sections:

Changes to This Document

This table lists the technical changes made to this document since it was first released.

Table 1: Changes to This Document

Date	Summary
April 2016	Initial release of this document.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you’re looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

CHAPTER 1

New and Changed BNG Features

This table summarizes the new and changed feature information for the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide*, and tells you where they are documented.

New and Changed Feature Information in Cisco IOS XR Release 6.0.x

Table 2: New and Changed Features in Cisco IOS XR Software

Feature	Description	Changed in Release	Where Documented
Activating IPv6 Router Advertisement on an IPv4 Subscriber Interface	This feature was introduced.	Release 6.0.1	*Configuring Subscriber Features* chapter Refer *BNG Neighbor Discovery Commands* chapter in *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*, for information on the commands used for ND configurations.
Linking to Subscriber Traffic in a Shared Policy Instance Group	This feature was introduced.	Release 6.0.1	Linking to Subscriber Traffic in a Shared Policy Instance Group, on page 238 topic in the *Configuring Subscriber Features* chapter

Feature	Description	Changed in Release	Where Documented
PPPoE LAC Subscriber Over PWHE	This feature was introduced.	Release 6.0.1	PPPoE LAC Subscriber Over PWHE, on page 172 topic in the *Establishing Subscriber Sessions* chapter

CHAPTER 2

Broadband Network Gateway Overview

This chapter provides an overview of the Broadband Network Gateway (BNG) functionality implemented on the Cisco ASR 9000 Series Router.

Table 3: Feature History for Broadband Network Gateway Overview

Release	Modification
Release 4.2.0	Initial release of BNG.
Release 5.3.3	RSP-880 support was added.

Understanding BNG

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 Provide (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 of subscriber sessions
- Address assignment

- Security
- Policy management
- Quality of Service (QoS)

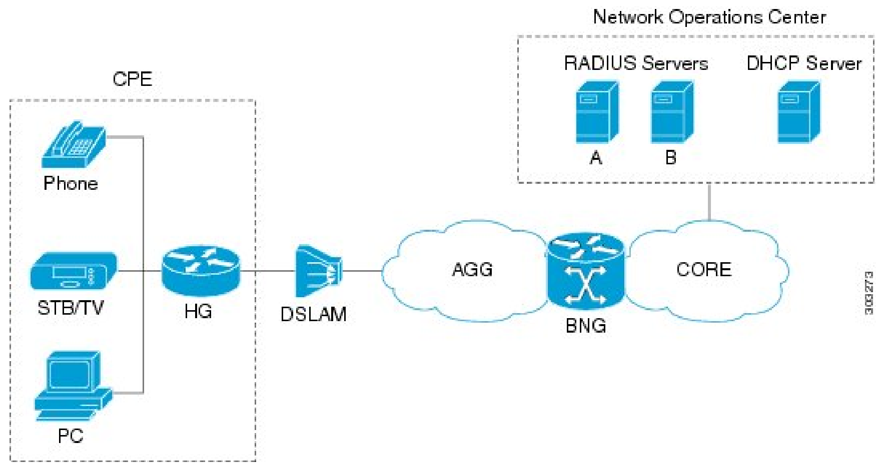
Some benefits of using BNG are:

- The BNG router not only performs the routing function but also communicates with authentication, authorization, and accounting (AAA) server to perform session management and billing functions. This makes the BNG solution more comprehensive.
- Different subscribers can be provided different network services. This enables the service provider to customize the broadband package for each customer based on their needs.

BNG Architecture

The goal of the BNG architecture is to enable the BNG router to interact with peripheral devices (like CPE) and servers (like AAA and DHCP), in order to provide broadband connectivity to subscribers and manage subscriber sessions. The basic BNG architecture is shown in this figure.

Figure 1: BNG Architecture



The BNG architecture is designed to perform these tasks:

- Connecting with the Customer Premise Equipment (CPE) that needs to be served broadband services.
- Establishing subscriber sessions using IPoE or PPPoE protocols.
- Interacting with the AAA server that authenticates subscribers, and keeps an account of subscriber sessions.
- Interacting with the DHCP server to provide IP address to clients.
- Advertising the subscriber routes.

The five BNG tasks are briefly explained in the following sections.

Connecting with the CPE

BNG connects to the CPE through a multiplexer and Home Gateway (HG). The CPE represents the triple play service in telecommunications, namely, voice (phone), video (set top box), and data (PC). The individual subscriber devices connect to the HG. In this example, the subscriber connects to the network over a Digital Subscriber Line (DSL) connection. Therefore, the HG connects into a DSL Access Multiplexer (DSLAM).

Multiple HGs can connect to a single DSLAM that sends the aggregated traffic to the BNG router. The BNG router routes traffic between the broadband remote access devices (like DSLAM or Ethernet Aggregation Switch) and the service provider network.

Establishing Subscriber Sessions

Each subscriber (or more specifically, an application running on the CPE) connects to the network by a logical session. Based on the protocol used, subscriber sessions are classified into two types:

- PPPoE subscriber session—The PPP over Ethernet (PPPoE) subscriber session is established using the point-to-point (PPP) protocol that runs between the CPE and BNG.
- IPoE subscriber session—The IP over Ethernet (IPoE) subscriber session is established using IP protocol that runs between the CPE and BNG; IP addressing is done using the DHCP protocol.

Interacting with the RADIUS Server

BNG relies on an external Remote Authentication Dial-In User Service (RADIUS) server to provide subscriber Authentication, Authorization, and Accounting (AAA) functions. During the AAA process, BNG uses RADIUS to:

- authenticate a subscriber before establishing a subscriber session
- authorize the subscriber to access specific network services or resources
- track usage of broadband services for accounting or billing

The RADIUS server contains a complete database of all subscribers of a service provider, and provides subscriber data updates to the BNG in the form of attributes within RADIUS messages. BNG, on the other hand, provides session usage (accounting) information to the RADIUS server. For more information about RADIUS attributes, see RADIUS Attributes, on page 339.

BNG supports connections with more than one RADIUS server to have fail over redundancy in the AAA process. For example, if RADIUS server A is active, then BNG directs all messages to the RADIUS server A. If the communication with RADIUS server A is lost, BNG redirects all messages to RADIUS server B.

During interactions between the BNG and RADIUS servers, BNG performs load balancing in a round-robin manner. During the load balancing process, BNG sends AAA processing requests to RADIUS server A only if it has the bandwidth to do the processing. Else, the request is send to RADIUS server B.

Interacting with the DHCP Server

BNG relies on an external Dynamic Host Configuration Protocol (DHCP) server for address allocation and client configuration functions. BNG can connect to more than one DHCP server to have fail over redundancy in the addressing process. The DHCP server contains an IP address pool, from which it allocates addresses to the CPE.

During the interaction between BNG and the DHCP server, BNG acts as a DHCP relay or DHCP proxy.

As the DHCP relay, BNG receives DHCP broadcasts from the client CPE, and forwards the request to the DHCP server.

As the DHCP proxy, BNG itself maintains the address pool by acquiring it from DHCP server, and also manages the IP address lease. BNG communicates on Layer 2 with the client Home Gateway, and on Layer 3 with the DHCP server.

The DSLAM modifies the DHCP packets by inserting subscriber identification information. BNG uses the identification information inserted by the DSLAM, as well as the address assigned by the DHCP server, to identify the subscriber on the network, and monitor the IP address lease.

Advertising Subscriber Routes

For optimal performance in design solutions where the Border Gateway Protocol (BGP) advertises the subscriber routes, the BNG advertises the entire subnet designated to the subscribers using the **network** command in the BGP configuration.

The BNG redistributes the individual subscriber routes only in scenarios where the Radius server assigns the IP address to a subscriber and there is no way to know to which BNG that particular subscriber will connect.

BNG Role in ISP Network Models

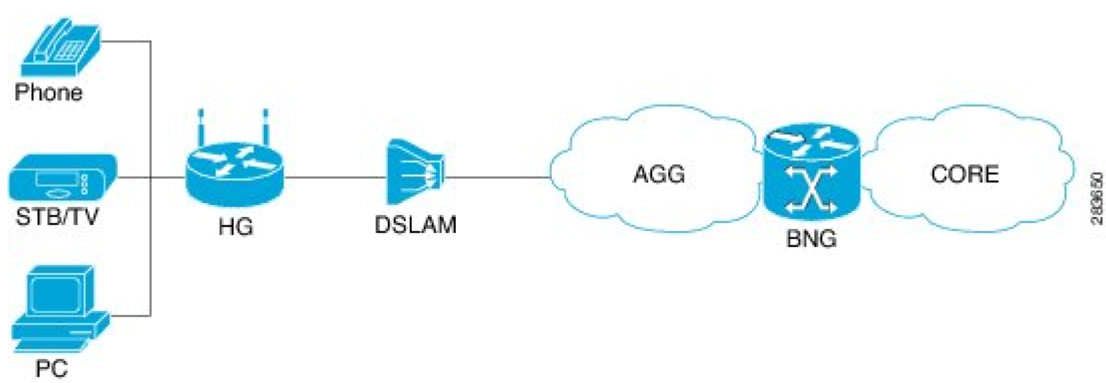
The role of BNG is to pass traffic from the subscriber to the ISP. The manner in which BNG connects to the ISP depends on the model of the network in which it is present. There are two types of network models:

- Network Service Provider, on page 6
- Access Network Provider, on page 7

Network Service Provider

The following figure shows the topology of a Network Service Provider model.

Figure 2: Network Service Provider Model

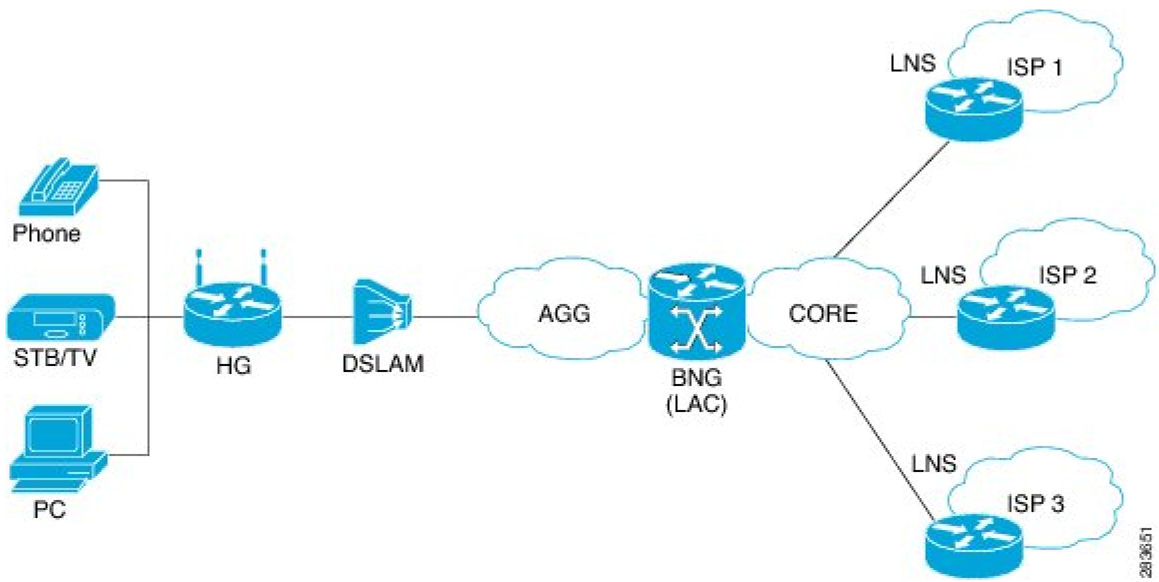


In the Network Service Provider model, the ISP (also called the retailer) directly provides the broadband connection to the subscriber. As shown in the above figure, BNG is at the edge router, and its role is to connect to the core network through uplinks.

Access Network Provider

The following figure shows the topology of a Access Network Provider model.

Figure 3: Access Network Provider Model



In the Access Network Provider model, a network carrier (also called the wholesaler) owns the edge network infrastructure, and provides the broadband connection to the subscriber. However, the network carrier does not own the broadband network. Instead, the network carrier connects to one of the ISPs that manage the broadband network.

BNG is implemented by the network carrier and its role is to hand the subscriber traffic off to one of several ISPs. The hand-off task, from the carrier to the ISP, is implemented by Layer 2 Tunneling Protocol (L2TP) or Layer 3 Virtual Private Networking (VPN). L2TP requires two distinct network components:

- L2TP Access Concentrator (LAC)—The LAC is provided by the BNG.
- L2TP Network Server (LNS)—The LNS is provided by the ISP.

BNG Packaging

The BNG pie, **asr9k-bng-px.pie** can be installed and activated on the Cisco ASR 9000 Series Router to access the BNG features. The install, uninstall, activate and deactivate operations can be performed without rebooting the router.

It is recommended that the relevant BNG configurations be removed from the running configuration of the router, before uninstalling or deactivating the BNG pie.

Installing and Activating the BNG Pie on Cisco ASR 9000 Series Router

Perform this task to install and activate the BNG pie on the Cisco ASR 9000 Series Router:

SUMMARY STEPS

1. **admin**
2. **install add** {*pie_location* | **source** | **tar**}
3. **install activate** {*pie_name* | **id**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters the administration mode.
Step 2	**install add** {*pie_location* | **source** | **tar**} **Example:** `RP/0/RSP0/CPU0:router(admin)# install add tftp://223.255.254.254/softdir/asr9k-bng-px.pie`	Installs the pie from the tftp location, on to the Cisco ASR 9000 Series Router.
Step 3	**install activate** {*pie_name* | **id**} **Example:** `RP/0/RSP0/CPU0:router(admin)# install activate asr9k-bng-px.pie`	Activates the installed pie on the Cisco ASR 9000 Series Router.

What to do next

Note During upgrade from Release 4.2.1 to Release 4.3.0, it is recommended that the Cisco ASR 9000 base image pie (asr9k-mini-px.pie) is installed prior to installing the BNG pie (asr9k-bng-px.pie).

After BNG pie is installed, you must copy BNG related configurations from the flash or tftp location to the router. If BNG pie is deactivated and activated again, then load the removed BNG configurations by executing the **load configuration removed** command from the configuration terminal.

Note Most of the BNG feature configurations are moved to a new namespace partition, and hence BNG features are not available by default now. To avoid inconsistent BNG configurations before, or after installing the BNG pie, run the **clear configuration inconsistency** command, in EXEC mode.

BNG Configuration Process

Configuring BNG on the Cisco ASR 9000 Series Router involves these stages:

- Configuring RADIUS Server—BNG is configured to interact with the RADIUS server for authentication, authorization, and accounting functions. For details, see Configuring Authentication, Authorization, and Accounting Functions, on page 13.
- Activating Control Policy—Control policies are activated to determine the action that BNG takes when specific events occur. The instructions for the action are provided in a policy map. For details, see Activating Control Policy, on page 61.
- Establishing Subscriber Sessions—Configurations are done to set up one or more logical sessions, from the subscriber to the network, for accessing broadband services. Each session is uniquely tracked and managed. For details, see Establishing Subscriber Sessions, on page 71.
- Deploying QoS—Quality of Service (QoS) is deployed to provide control over a variety of network applications and traffic types. For example, the service provider can have control over resources (example bandwidth) allocated to each subscriber, provide customized services, and give priority to traffic belonging to mission-critical applications. For details, see Deploying the Quality of Service (QoS), on page 175.
- Configuring Subscriber Features—Configurations are done to activate certain subscriber features that provide additional capabilities like policy based routing, access control using access list and access groups, and multicast services. For details, see Configuring Subscriber Features, on page 219.
- Verifying Session Establishment—Established sessions are verified and monitored to ensure that connections are always available for use. The verification is primarily done using "show" commands. Refer to the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference* guide for the list of various "show" commands.

To use a BNG command, you must be in a user group associated with a task group that includes the proper task IDs. The *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference* guide includes the task IDs required for each command. If you suspect that the user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Restriction

The Select VRF Download (SVD) must be disabled, when BNG is configured. For more information about SVD, see the *Cisco IOS XR Routing Configuration Guide for the Cisco XR 12000 Series Router*.

Hardware Requirements for BNG

These hardwares support BNG:

- BNG is supported on Satellite Network Virtualization (nV) system.
- BNG is supported on Cisco ASR 9000 Series Aggregation Services Routers only with RSP-440, RSP-880 route switch processors.

Table 4: Line Cards and Modular Port Adapters Supported on BNG

Product Description	Part Number
24-Port 10-Gigabit Ethernet Line Card, Service Edge Optimized	A9K-24X10GE-SE
36-Port 10-Gigabit Ethernet Line Card, Service Edge Optimized	A9K-36X10GE-SE

Product Description	Part Number
40-Port Gigabit Ethernet Line Card, Service Edge Optimized	A9K-40GE-SE
4-Port 10-Gigabit Ethernet, 16-Port Gigabit Ethernet Line Card, 40G Service Edge Optimized	A9K-4T16GE-SE
80 Gigabyte Modular Line Card, Service Edge Optimized	A9K-MOD80-SE
160 Gigabyte Modular Line Card, Service Edge Optimized	A9K-MOD160-SE
20-Port Gigabit Ethernet Modular Port Adapter (MPA)	A9K-MPA-20GE
2-port 10-Gigabit Ethernet Modular Port Adapter (MPA)	A9K-MPA-2X10GE
4-Port 10-Gigabit Ethernet Modular Port Adapter (MPA)	A9K-MPA-4X10GE
2-port 40-Gigabit Ethernet Modular Port Adapter (MPA)	A9K-MPA-2X40GE
1-Port 40-Gigabit Ethernet Modular Port Adapter (MPA)	A9K-MPA-1X40GE

BNG Interoperability

The BNG interoperability allows BNG to exchange and use information with other larger heterogeneous networks. These are the key features:

- BNG Coexists with ASR9001:

  ASR9001 is a standalone high processing capability router that comprises of a route switch processor (RSP), linecards (LC), and ethernet plugs (EPs). All BNG features are fully supported on the ASR9001 chassis.

- BNG Supports nV Satellite:

  The only topology that is supported with BNG-nV Satellite is - bundled Ethernet ports on the CPE side of the Satellite node connected to the Cisco ASR 9000 through non-bundle configuration (static-pinning). That is,

  CPE --- Bundle --- [Satellite] --- Non Bundle ICL --- ASR9K

  Although the following topology is supported on Satellite nV System (from Cisco IOS XR Software Release 5.3.2 onwards), it is not supported on BNG:

  - Bundled Ethernet ports on the CPE side of the satellite node, connected to the Cisco ASR 9000 through bundle Ethernet connection.

- BNG interoperates with Carrier Grade NAT (CGN):

  To address the impending threat from IPv4 address space depletion, it is recommended that the remaining or available IPv4 addresses be shared among larger numbers of customers. This is done by using CGN, which primarily pulls the address allocation to a more centralized NAT in the service provider network. NAT44 is a technology that uses CGN and helps manage depletion issues of the IPv4 address space. BNG supports the ability to perform NAT44 translation on IPoE and PPPoE-based BNG subscriber sessions.

  Note For BNG and CGN interoperability, configure the BNG interface and the application service virtual interface (SVI) on the same VRF instance.

Restrictions

- Only bundle access with non-bundle ICLs are supported for BNG interfaces over Satellite nV System access interfaces.

BNG Smart Licensing

BNG supports Cisco Smart Software Licensing that provides a simplified way for the customers to purchase licenses and to manage them across their network. This provides a customizable consumption-based model that aligns to the network growth of the customer. It also provides the flexibility to quickly modify or upgrade software feature configurations to deploy new services over time.

For more information about Cisco Smart Software Licensing, see Software Entitlement on the Cisco ASR 9000 Series Router chapter of *System Management Configuration Guide for Cisco ASR 9000 Series Routers*.

For latest updates, refer the latest version of guides present in http://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xr-software/products-installation-and-configuration-guides-list.html.

BNG Smart Licensing supports Geo redundancy as well as non-Geo redundancy subscriber sessions. One license is required for every group of 8000 subscribers or a fraction of it. For example, two licenses are required for 9000 subscribers.

These are the software license PIDs for BNG:

- S-A9K-BNG-LIC-8K —for non-geo redundancy sessions
- S-A9K-BNG-ADV-8K —for geo redundancy sessions

You can use the **show sessionmon license** command to display the subscriber session statistics.

CHAPTER 3

Configuring Authentication, Authorization, and Accounting Functions

This chapter provides information about configuring authentication, authorization, and accounting (AAA) functions on the BNG router. 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), and Service Accounting for QoS.

Table 5: Feature History for Configuring Authentication, Authorization, and Accounting Functions

Release	Modification
Release 4.2.0	Initial release
Release 5.3.1	RADIUS over IPv6 was introduced.
Release 5.3.2	Service accounting support was added for line card subscribers.
Release 6.2.1	A new MAC address format was introduced for RADIUS User-name Attribute.

This chapter covers these topics:

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. If the sessions on the BNG are not acknowledged for their Account-Start request, loss of sessions on route processor fail over (RPFO) and other critical failures are reported. 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.

The **radius-server deadtime** *limit* command can be used to configure the **deadtime** for RADIUS server. For details, see Configuring RADIUS Server Settings, on page 30.

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 15.

Configuring RADIUS Server Group

Perform this task to define a named server group as the server host.

SUMMARY STEPS

1. **configure**
2. **aaa group server radius** *name*
3. **accounting accept** *radius_attribute_list_name*
4. **authorization reply accept** *radius_attribute_list_name*
5. **deadtime** *limit*
6. **load-balance method least-outstanding batch-size** *size* **ignore-preferred-server**
7. **server** *host_name* **acct-port** *accounting_port_number* **auth-port** *authentication_port_number*

8. **source-interface** *name* *value*
9. **vrf** *name*
10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**aaa group server radius** *name* **Example:** `RP/0/RSP0/CPU0:router(config)# aaa group server radius r1`	Configures the RADIUS server group named r1.
Step 3	**accounting accept** *radius_attribute_list_name* **Example:** `RP/0/RSP0/CPU0:router(config-sg-radius)# accounting accept att_list`	Configures the radius attribute filter for the accounting process to accept only the attributes specified in the list.
Step 4	**authorization reply accept** *radius_attribute_list_name* **Example:** `RP/0/RSP0/CPU0:router(config-sg-radius)# authorization reply accept att_list1`	Configures the radius attribute filter for the authorization process to accept only the attributes specified in the list.
Step 5	**deadtime** *limit* **Example:** `RP/0/RSP0/CPU0:router(config-sg-radius)# deadtime 40`	Configures the RADIUS server-group deadtime. The deadtime limit is configured in minutes. The range is from 1 to 1440, and the default is 0.
Step 6	**load-balance method least-outstanding batch-size** *size* **ignore-preferred-server** **Example:** `RP/0/RSP0/CPU0:router(config-sg-radius)# load-balance method least-outstanding batch-size 50 ignore-preferred-server`	Configures load balancing batch size after which the next host is picked.
Step 7	**server** *host_name* **acct-port** *accounting_port_number* **auth-port** *authentication_port_number* **Example:** `RP/0/RSP0/CPU0:router(config-sg-radius)# server 1.2.3.4 acct-port 455 auth-port 567`	Specifies the radius server, and its IP address or host name. Configures the UDP port for RADIUS accounting and authentication requests. The accounting and authentication port number ranges from 0 to 65535. If no value is specified, then the default is 1645 for auth-port, and 1646 for acct-port. From Cisco IOS XR Software Release 5.3.1 and later, IPv6 address can also be configured for the RADIUS server.

	Command or Action	Purpose
		But, the host name option is supported only for IPv4 domain, and not for IPv6.
Step 8	**source-interface** *name value* **Example:** `RP/0/RSP0/CPU0:router(config-sg-radius)# source-interface Bundle-Ether 455`	Configures the RADIUS server-group source-interface name and value for Bundle-Ether.
Step 9	**vrf** *name* **Example:** `RP/0/RSP0/CPU0:router(config-sg-radius)# vrf vrf_1`	Configures the vrf to which the server radius group belongs.
Step 10	**commit**	

Configuring Radius Server-Group: An example

```
configure
aaa group server radius r1
accounting accept r1 r2
authorization reply accept a1 a2
deadtime 8
load-balance method least-outstanding batch-size 45 ignore-preferred-server
server host_name acct-port 355 auth-port 544
source-interface Bundle-Ether100.10
vrf vrf_1
!
end
```

Specifying Method List

Method lists for AAA define 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 list must be applied to the configuration mechanism responsible for validating user-access credentials. The only exception to this requirement is the default method list (named "default"). The default method list is automatically applied if no other method list is defined. A defined method list overrides the default method list.

On BNG, you have to specify the method list and the server group that will be used for AAA services. For specifying method lists, see Configuring Method Lists for AAA, on page 17.

Configuring Method Lists for AAA

Perform this task to assign the method list to be used by the server group for subscriber authentication, authorization, and accounting.

SUMMARY STEPS

1. **configure**
2. **aaa authentication subscriber default** *method-list-name* **group** *server-group-name*
3. **aaa authorization subscriber default** *method-list-name* **group** *server-group-name* | **radius**
4. **aaa accounting subscriber default** *method-list-name* **group** *server-group-name*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**aaa authentication subscriber default** *method-list-name* **group** *server-group-name* **Example:** `RP/0/RSP0/CPU0:router(config)# aaa authentication subscriber default method1 group group1 radius group group2 group group3 ...`	Configures the method-list which will be applied by default for subscriber authentication. You can either enter 'default' or a user-defined name for the AAA method-list. Also, enter the name of the server group, on which the method list is applied.
Step 3	**aaa authorization subscriber default** *method-list-name* **group** *server-group-name* | **radius** **Example:** `RP/0/RSP0/CPU0:router(config)# aaa authorization subscriber default method1 group group1 radius group group2 group group3 ...`	Configures the method-list which will be applied by default for subscriber authorization. You can either enter 'default' or a user-defined name for the AAA method-list. Also, enter the name of the server group, on which the method list is applied.
Step 4	**aaa accounting subscriber default** *method-list-name* **group** *server-group-name* **Example:** `RP/0/RSP0/CPU0:router(config)# aaa accounting subscriber default method1 group group1 radius group group2 group group3 ...`	Configures the method-list which will be applied by default for subscriber accounting. You can either enter 'default' or a user-defined name for the AAA method-list. Also, enter the name of the server group, on which the method list is applied.
Step 5	**commit**	

Configuring Method-list for AAA: An example

```
configure
aaa authentication subscriber default group radius group rad2 group rad3..
aaa authorization subscriber default group radius group rad1 group rad2 group rad3..
aaa accounting subscriber default group radius group rad1 group rad2 group rad3..
!
!
end
```

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 339.

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 List

An attribute list is named list that contains a set of attributes. You can configure the RADIUS server to use a particular attribute list to perform the AAA function.

To create an attribute list, see Configuring RADIUS Attribute List, on page 25.

AAA Attribute Format

It is possible to define a customized format for some attributes. The configuration syntax for creating a new format is:

```
aaa attribute format <format-name> format-string [length] <string> *[<Identity-Attribute>]
```

where:

- format-name — Specifies the name given to the attribute format. This name is referred when the format is applied on an attribute.
- length — (Optional) Specifies the maximum length of the formatted attribute string. If the final length of the attribute string is greater than the value specified in LENGTH, it is truncated to LENGTH bytes. The maximum value allowed for LENGTH is 255. If the argument is not configured, the default is also 255.
- string — Contains regular ASCII characters that includes conversion specifiers. Only the % symbol is allowed as a conversion specifier in the STRING. The STRING value is enclosed in double quotes.
- Identity-Attribute — Identifies a session, and includes user-name, ip-address, and mac-address. A list of currently-defined identity attributes is displayed on the CLI.

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 20.

To create a customized nas-port attribute and apply a predefined format to nas-port-ID attribute , see Configuring RADIUS Attribute Format, on page 26.

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 Attribute Format Function, on page 28.

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

An example of constructing a username in the form "mac-address@vendor-class-ID" is:

```
aaa attribute format USERNAME-FORMAT format-string "%s@%s" mac-address dhcp-vendor-class
```

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). The extended nas-port-ID command is:

```
aaa radius attribute nas-port-id format <FORMAT_NAME> [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. An example of constructing a maximum 128 byte NAS-Port-ID, by combining the BNG port information and Circuit-ID is:

```
aaa attribute format NAS-PORT-ID-FORMAT1 format-string length 128 "eth %s/%s/%s:%s.%s %s"
physical-slot physical-subslot physical-port outer-vlan-Id inner-vlan-id circuit-id-tag
```

An example of constructing the NAS-Port-ID from just the BNG port information, and with "0/0/0/0/0/0" appended at the end for circuit-ID, is:

```
aaa attribute format NAS-PORT-ID-FORMAT2 format-string "eth %s/%s/%s:%s.%s 0/0/0/0/0/0"
physical-slot physical-subslot physical-port outer-vlan-Id inner-vlan-id
```

An example of constructing the NAS-Port-ID from just the Circuit-ID is:

```
aaa attribute format NAS-PORT-ID-FORMAT3 format-string "%s" circuit-id-tag
```

The NAS-Port-ID formats configured in the above examples, can be specified in the nas-port-ID command, thus:

```
For  IPoEoQINQ interface:-
aaa radius attribute  nas-port-id format NAS-PORT-ID-FORMAT1  type 41

For  Virtual IPoEoQINQ interface:-
aaa radius attribute  nas-port-id format NAS-PORT-ID-FORMAT2 type 44

For  IPOEoE interface:-
aaa radius attribute  nas-port-id format NAS-PORT-ID-FORMAT3 type 39
```

NAS-Port-Type on Interface or VLAN Sub-interface

In order to have different production models for subscribers on the same BNG router, but different physical interfaces of same type, the NAS-Port-Type is made configurable for each physical interface, or VLAN sub-interface. With a different NAS-Port-Type value configured on the interface, the NAS-Port and NAS-Port-ID gets formatted according to the formats defined globally for the new NAS-Port-Type configured on the interface, instead of the actual value of NAS-Port-Type that the interface has. This in turn sends different formats of NAS-Port, NAS-Port-ID and NAS-Port-Type to the RADIUS server for the subscribers under different production models.

In the case of sub-interfaces, the hierarchy to be followed in deciding the format of NAS-Port-Type to be sent to the RADIUS server is:

1. Verify whether the NAS-Port-Type is configured on the sub-interface in which the subscriber session arrives.
2. If NAS-Port-Type is not configured on the sub-interface, verify whether it is configured on the main physical interface.

   The format of NAS-Port or NAS-Port-ID is based on the NAS-Port-Type retrieved in Step 1 or Step 2.
3. If NAS-Port-Type is configured on neither the sub-interface nor the main physical interface, the format of NAS-Port or NAS-Port-ID is based on the format of the default NAS-Port-Type of the sub-interface.
4. If a NAS-Port or NAS-Port-ID format is not configured for the NAS-Port-Type retrieved in steps 1, 2 or 3, the format of NAS-Port or NAS-Port-ID is based on the default formats of NAS-Port or NAS-Port-ID.

Use this command to configure NAS-Port-Type per interface or VLAN sub-interface:

```
aaa radius attribute nas-port-type <nas-port-type>
```

where:

<nas-port-type> is either a number ranging from 0 to 44, or a string specifying the nas-port-type.

Refer Configuring RADIUS Attribute Nas-port-type, on page 27.

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.

The command used to configure the calling-station-ID and called-station-ID attributes is:

```
aaa radius attribute calling-station-id format <FORMAT_NAME>

aaa radius attribute called-station-id format <FORMAT_NAME>
```

Examples of constructing calling-station-ID from mac-address, remote-ID, and circuit-ID are:

```
aaa radius attribute calling-station-id format CLID-FORMAT

aaa attribute format CLID-FORMAT format-string “%s:%s:%s” client-mac-address-ietf
remote-id-tag circuit-id-tag
```

Examples of constructing called-station-ID from mac-address, remote-ID, and circuit-ID are:

```
aaa radius attribute called-station-id format CLDID-FORMAT

aaa attribute format CLDID-FORMAT format-string "%s:%s" client-mac-address-raw circuit-id-tag
```

NAS-Port Format

NAS-Port is a 4-byte value that has the physical port information of the Broadband Remote Access Server (BRAS), which connects the Access Aggregation network to BNG. It is used both by Access-Request packets and Accounting-Request packets. To uniquely identify a physical port on BRAS, multiple pieces of information such as shelf, slot, adapter, and so on is used along with the port number. A configurable format called format-e is defined to allow individual bits or group of bits in 32 bits of NAS-Port to represent or encode various pieces that constitute port information.

Individual bits in NAS-Port can be encoded with these characters:

- Zero: 0
- One: 1
- PPPoX slot: S
- PPPoX adapter: A
- PPPoX port: P
- PPPoX VLAN Id: V
- PPPoX VPI: I
- PPPoX VCI: C
- Session-Id: U
- PPPoX Inner VLAN ID: Q

```
aaa radius attribute nas-port format e [string] [type {nas-port-type}]
```

The above command is used to configure a format-e encode string for a particular interface of NAS-Port type (RADIUS attribute 61). The permissible nas-port type values are:

Nas-port-types	Values	Whether value can be derived from associated interface	Whether value can be configured on the interface configuration mode
ASYNC	0	No	Yes
SYNC	1	No	Yes
ISDN	2	No	Yes
ISDN_V120	3	No	Yes
ISDN_V110	4	No	Yes

Nas-port-types	Values	Whether value can be derived from associated interface	Whether value can be configured on the interface configuration mode
VIRTUAL	5	No	Yes
ISDN_PIAFS	6	No	Yes
X75	9	No	Yes
ETHERNET	15	No	Yes
PPPATM	30	No	Yes
PPPOEOA	31	No	Yes
PPPOEOE	32	Yes	Yes
PPPOEOVLAN	33	Yes	Yes
PPPOEOQINQ	34	Yes	Yes
VIRTUAL_PPPOEOE	35	Yes	Yes
VIRTUAL_PPPOEOVLAN	36	Yes	Yes
VIRTUAL_PPPOEOQINQ	37	Yes	Yes
IPSEC	38	No	Yes
IPOEOE	39	Yes	Yes
IPOEOVLAN	40	Yes	Yes
IPOEOQINQ	41	Yes	Yes
VIRTUAL_IPOEOE	42	Yes	Yes
VIRTUAL_IPOEOVLAN	43	Yes	Yes
VIRTUAL_IPOEOQINQ	44	Yes	Yes

Examples:

```
For non-bundle: GigabitEthernet0/1/2/3.11.pppoe5

where:
PPPoEoQinQ (assuming 2 vlan tags): interface-type
1: slot
2: adapter
3: port
vlan-ids: whatever the outer and inner vlan-ids received in the PADR were
5: session-id

aaa radius attribute nas-port format e SSAAPPPPQQQQQQQQQQVVVVVVVVVVUUUU type 34
Generated NAS-Port:     01100011QQQQQQQQQQVVVVVVVVVV0101
```

```
For bundle: Bundle-Ether17.23.pppoe8
where:
Virtual-PPPoEoQinQ (assuming 2 vlan tags): interface-type
0: slot
0: adapter
17 (bundle-id): port
Vlan-Ids: whatever the outer and inner vlan-ids received in the PADR were.
8: session-id

aaa radius attribute nas-port format e PPPPPPQQQQQQQQQQVVVVVVVVVVUUUUUU type 37
Generated NAS-Port:     010001QQQQQQQQQQVVVVVVVVVV000101
```

NAS-port format for IP/DHCP sessions are represented in these examples:

```
For IPoEoVLAN interface type:
aaa radius attribute nas-port format e SSAAAPPPPPVVVVVVVVVVVVVVVVVVVVVV type 40

For IPoEoQinQ:
aaa radius attribute nas-port format e SSAAAPPPPPQQQQQQQQQQQVVVVVVVVVVV type 41

For virtual IPoEoVLAN:
aaa radius attribute nas-port format e PPPPPPPPVVVVVVVVVVVVVVVVUUUUUUUU type 43
```

NAS-port format for PPPoE sessions are represented in these examples:

```
For PPPoEoVLAN  interface type:
aaa radius attribute nas-port format e SSAAAPPPPPVVVVVVVVVVVVVVVVVVUUUU type 33

For Virtual PPPoEoVLAN:.
aaa radius attribute nas-port format e PPPPPPPPVVVVVVVVVVVVVVVVUUUUUUUU type 36
```

Note If a NAS-Port format is not configured for a NAS-Port-Type, the system looks for a default CLI configuration for the NAS-Port format. In the absence of both these configurations, for sessions with that particular NAS-Port-Type, the NAS-Port attribute is not sent to the RADIUS server.

Configuring RADIUS Attribute List

Perform this task to create a RADIUS attribute list that is used for filtering authorization and accounting attributes.

SUMMARY STEPS

1. **configure**
2. **radius-server attribute list** *listname*
3. **attribute** *list_of_radius_attributes*
4. **attribute vendor-id** *vendor-type number*
5. **vendor-type** *vendor-type-value*
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**radius-server attribute list** *listname* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server attribute list l1`	Defines the name of the attribute list.
Step 3	**attribute** *list_of_radius_attributes* **Example:** `RP/0/RSP0/CPU0:router(config-attribute-filter)# attribute a1, a2`	Populates the list with radius attributes. **Note** For more information about supported attributes, see RADIUS Attributes, on page 339 .
Step 4	**attribute vendor-id** *vendor-type number* **Example:** `RP/0/RSP0/CPU0:router(config)# attribute vendor-id 6456`	Configures the attribute filtering to be applied to vendor specific attributes (VSAs) by allowing vendor specific information for VSAs to be specified in radius attribute list CLI. Vendor specific information comprises of vendor-id, vendor-type, and optional attribute name in case of Cisco generic VSA. The vendor-id ranges from 0 to 4294967295.
Step 5	**vendor-type** *vendor-type-value* **Example:** `RP/0/RSP0/CPU0:router(config-attribute-filter-vsa)# vendor-type 54`	Configures the vendor specific information such as the vendor-type to be specified in radius attribute list. The range of the vendor-type value is from 1 to 254.
Step 6	**commit**	

Configuring RADIUS Attribute List: An example

```
configure
radius-server attribute list list_! attribute B C
attribute vendor-id vendor-type 10
vendor-type 30
!
end
```

Configuring RADIUS Attribute Format

Perform this task to the define RADIUS attribute format for the nas-port attribute, and apply a predefined format on nas-port-ID attribute.

SUMMARY STEPS

1. **configure**
2. **aaa radius attribute**

3. **nas-port format e** *string* **type** *nas-port-type value*
4. **nas-port-id format** *format name*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**aaa radius attribute** **Example:** `RP/0/RSP0/CPU0:router(config)# aaa radius attribute`	Configures the AAA radius attribute.
Step 3	**nas-port format e** *string* **type** *nas-port-type value* **Example:** `RP/0/RSP0/CPU0:router(config)# nas-port format e format1 type 30`	Configures the format for nas-port attribute. The string represents a 32 character string representing the format to be used. The nas-port-value ranges from 0 to 44.
Step 4	**nas-port-id format** *format name* **Example:** `RP/0/RSP0/CPU0:router(config)# nas-port-id format format2`	Applies a predefined format to the nas-port-ID attribute.
Step 5	**commit**	

Configuring RADIUS Attribute Format: An example

```
configure
aaa radius attribute
nas-port format e abcd type 40
nas-port-id format ADEF
!
end
```

Configuring RADIUS Attribute Nas-port-type

Perform this task to configure RADIUS Attribute nas-port-type on a physical interface or VLAN sub-interface:

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-name*
3. **aaa radius attribute nas-port-type** {*value* | *name*}
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-name* **Example:** `RP/0/RSP0/CPU0:router(config)# interface gigabitEthernet 0/0/0/0`	Enters the interface configuration mode.

Step 3	**aaa radius attribute nas-port-type** {*value* | *name*}
	Example:
	`RP/0/RSP0/CPU0:router(config-if)# aaa radius attribute nas-port-type 30`
	or
	`RP/0/RSP0/CPU0:router(config-if)# aaa radius attribute nas-port-type Ethernet`
		Configures the RADIUS Attribute nas-port-type value.
		The range of *value* is from 0 to 44.
		See table in NAS-Port Format, on page 23, for permissible nas-port-type values within this range.

Step 4	**commit**	

Configuring RADIUS Attribute Nas-port-type: An example

```
configure
interface gigabitEthernet 0/0/0/0
 aaa radius attribute nas-port-type Ethernet
!
end
```

Configuring AAA Attribute Format Function

Perform this task to configure a function for the AAA attribute format. The function is for stripping the user-name till the delimiter.

SUMMARY STEPS

1. **configure**
2. **aaa attribute format** *format-name*
3. **username-strip prefix-delimiter** *prefix_delimiter*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**aaa attribute format** *format-name* **Example:** `RP/0/RSP0/CPU0:router(config)# aaa attribute format red`	Specifies the format name for which the function is defined.
Step 3	**username-strip prefix-delimiter** *prefix_delimiter* **Example:** `RP/0/RSP0/CPU0:router(config-id-format)# username-strip prefix-delimiter @`	Configures the function to strip the username preceding the prefix delimiter, which is @.
Step 4	**commit**	

Configuring AAA Attribute Format Function: An example

```
configure
aaa attribute format red
username-strip prefix-delimiter @
!
!
end
```

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. This table lists some of the key settings:

Settings	Description
Server host	Defines the RADIUS server details to which BNG will connect.
Attribute list	Defines which attribute list is to be used.
Server key	Defines the encryption status.
Dead criteria	Defines the criteria that is used to mark a RADIUS server as dead.
Retransmit value	Defines the number of retries the BNG makes to send data to RADIUS server.
Timeout value	Defines how long BNG waits for the RADIUS server to reply.
Automated testing	Defines the duration after which automated testing will start and the username to be tested.
IP DSCP	Allows RADIUS packets to be marked with a specific Differentiated Services Code Point (DSCP) value.

For more making RADIUS server settings, see Configuring RADIUS Server Settings, on page 30.

For more making specific automated testing settings, see Configuring Automated Testing, on page 33.

For more making specific IP DSCP settings, see Setting IP DSCP for RADIUS Server, on page 34.

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.

Configuring RADIUS Server Settings

Perform this task to make RADIUS server specific settings on the BNG router.

SUMMARY STEPS

1. **configure**
2. **radius-server host** *ip-address* **acct-port** *accounting_port_number* **auth-port** *authentication_port_number*
3. **radius-server attribute list** *list_name attribute_list*
4. **radius-server key** *7 encrypted_text*
5. **radius-server disallow null-username**
6. **radius-server dead-criteria time** *value*
7. **radius-server dead-criteria tries** *value*
8. **radius-server deadtime** *limit*
9. **radius-server ipv4 dscp** *codepoint_value*
10. **radius-server load-balance method least-outstanding ignore-preferred-server batch-size** *size*
11. **radius-server retransmit** *retransmit_value*
12. **radius-server source-port extended**
13. **radius-server timeout** *value*
14. **radius-server vsa attribute ignore unknown**
15. **radius source-interface Loopback** *value* **vrf** *vrf_name*
16. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**radius-server host** *ip-address* **acct-port** *accounting_port_number* **auth-port** *authentication_port_number* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server host 1.2.3.4 acct-port 455 auth-port 567`	Specifies the radius server and its IP address. Configures the UDP port for RADIUS accounting and authentication requests. The accounting and authentication port numbers range from 0 to 65535. If no value is specified, then the default is 1645 for the auth-port and 1646 for the acct-port. From Cisco IOS XR Software Release 5.3.1 and later, IPv6 address can also be configured for the RADIUS server host.
Step 3	**radius-server attribute list** *list_name attribute_list* **Example:**	Specifies the radius server attributes list, and customizes the selected radius attributes.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# radius-server attribute list rad_list a b`	
Step 4	**radius-server key** *7 encrypted_text* **Example:** `RP/0/RSP0/CPU0:router(config-radius-host)# radius-server key 7 rngiry`	Specifies the per-server encryption key that overrides the default, and takes the value 0 or 7, which indicates that the unencrypted key will follow.
Step 5	**radius-server disallow null-username** **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server disallow null-username`	Specifies that the null-username is disallowed for the radius server.
Step 6	**radius-server dead-criteria time** *value* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server dead-criteria time 40`	Specifies the dead server detection criteria for a configured RADIUS server. The time (in seconds) specifies the minimum time that must elapse since a response is received from this RADIUS server.
Step 7	**radius-server dead-criteria tries** *value* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server dead-criteria tries 50`	Specify the value for the number of consecutive timeouts that must occur on the router before the RADIUS server is marked as dead. The value ranges from 1 to 100.
Step 8	**radius-server deadtime** *limit* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server deadtime 67`	Specifies the time in minutes for which a RADIUS server is marked dead. The deadtime limit is specified in minutes and ranges from 1 to 1440. If no value is specified, the default is 0.
Step 9	**radius-server ipv4 dscp** *codepoint_value* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server ipv4 dscp 45`	Allows radius packets to be marked with a specific differentiated services code point (DSCP) value. This code point value ranges from 0 to 63.
Step 10	**radius-server load-balance method least-outstanding ignore-preferred-server batch-size** *size* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server load-balance method least-outstanding ignore-preferred-server batch-size 500`	Configures the radius load-balancing options by picking the server with the least outstanding transactions. This load-balancing method uses the batch-size for the selection of the server. The size ranges from 1 to 1500. If no value is specified, the default is 25.
Step 11	**radius-server retransmit** *retransmit_value* **Example:**	Specifies the number of retries to the active server. The retransmit value indicates the number of retries in numeric

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# radius-server retransmit 45`	and ranges from 1 to 100. If no value is specified, then the default is 3.
Step 12	**radius-server source-port extended** **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server source-port extended`	Configures BNG to use a total of 200 ports as the source ports for sending out RADIUS requests.
Step 13	**radius-server timeout** *value* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server timeout`	Specifies the time to wait for a radius server to reply. The value is in seconds and ranges from 1 to 1000. The default is 5.
Step 14	**radius-server vsa attribute ignore unknown** **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server vsa attribute ignore unknown`	Ignores the unknown vendor-specific attributes for the radius server.
Step 15	**radius source-interface Loopback** *value* **vrf** *vrf_name* **Example:** `RP/0/RSP0/CPU0:router(config)# radius source-interface Loopback 655 vrf vrf_1`	Specifies loopback interface for source address in RADIUS packets. The value ranges from 0 to 65535.
Step 16	**commit**	

Configuring RADIUS Server Settings: Examples

```
\\Configuring RADIUS Server Options
configure
radius-server attribute list list1 a b
radius-server dead-criteria time 100
radius-server deadtime 30
radius-server disallow null-username
radius-server host 1.2.3.4 acct-port 655 auth-port 566
radius-server ipv4 dscp 34
radius-server key 7 ERITY$
radius-server load-balance method least-outstanding ignore-preferred-server batch-size 25
radius-server retransmit 50
radius-server source-port extended
radius-server timeout 500
radius-server vsa attribute ignore unknown
!
!
end

\\Configuring RADIUS Attribute List
radius-server attribute list list_! attribute B C
attribute vendor-id vendor-type 10
vendor-type 30
```

```
!
end

\\Configuring RADIUS Server Host
configure
radius-server host 1.3.5.7 acct-port 56 auth-port 66
idle-time 45
ignore-acct-port
ignore-auth-port 3.4.5.6
key 7 ERWQ
retransmit 50
test username username
timeout 500
!
end

\\Configuring RADIUS Server Key
configure
radius-server key 7  ERWQ
!
end

\\Configuring Load Balancing for RADIUS Server
configure
radius-server load-balance method least-outstanding batch-size 25
radius-server load-balance method least-outstanding ignore-preferred-server batch-size 45
!
end

\\Ignoring Unknown VSA Attributes in RADIUS Server
configure
radius-server vsa attribute ignore unknown
!
end

\\Configuring Dead Criteria for RADIUS Server
configure
radius-server dead-criteria time 60
radius-server dead-criteria tries 60
!
end

\\Configuring Disallow Username
configure
radius-server disallow null-username
!
end

\\Setting IP DSCP for RADIUS Server
configure
radius-server ipv4 dscp 43
radius-server ipv4 dscp default
!
end
```

Configuring Automated Testing

Perform this task to test if the external RADIUS server is UP or not.

SUMMARY STEPS

1. **configure**
2. **radius-server idle-time** *idle_time*
3. **radius-server test username** *username*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**radius-server idle-time** *idle_time* **Example:** `RP/0/RSP0/CPU0:router(config-radius-host)# radius-server idle-time 45`	Specifies the idle-time after which the automated test should start. The idle time is specified in minutes, and ranges from 1 to 60.
Step 3	**radius-server test username** *username* **Example:** `RP/0/RSP0/CPU0:router(config-radius-host)# radius-server test username user1`	Specifies the username to be tested for the automated testing functionality.
Step 4	**commit**	

Configuring Automated Testing: An example

```
configure
radius-server idle-time 60
radius-server test username user_1
!
end
```

Setting IP DSCP for RADIUS Server

Perform this task to set IP differentiated services code point (DSCP) for RADIUS server.

SUMMARY STEPS

1. **configure**
2. **radius-server ipv4 dscp** *codepoint_value*
3. **radius-server ipv4 dscp default**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**radius-server ipv4 dscp** *codepoint_value* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server ipv4 dscp 45`	Allows radius packets to be marked with a specific differentiated services code point (DSCP) value that replaces the outdated IP precedence, a 3-bit field in the Type of Service byte of the IP header originally used to classify and prioritize types of traffic. This code point value ranges from 0 to 63.
Step 3	**radius-server ipv4 dscp default** **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server ipv4 dscp default`	Matches the packets with default dscp (000000).
Step 4	**commit**	

Setting IP DSCP for RADIUS Server: An example

```
configure
radius-server ipv4 dscp 43
radius-server ipv4 dscp default
!
end
```

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.

You can configure the load balancing feature either globally, or for RADIUS servers that are part of a server group. In the server group, if a preferred server is defined, you need to include the keyword "ignore-preferred-server" in the load-balancing configuration, to disable the preference.

For configuring the load balancing feature globally, see Configuring Load Balancing for Global RADIUS Server Group, on page 36.

For configuring the load balancing feature on RADIUS servers that are part of a named server group, see Configuring Load Balancing for a Named RADIUS Server Group, on page 36.

Configuring Load Balancing for Global RADIUS Server Group

Perform this task to activate the load balancing function for the global RADIUS server group. As an example, in this configuration the preferred server is set to be ignored.

SUMMARY STEPS

1. **configure**
2. **radius-server load-balance method least-outstanding batch-size** *size*
3. **radius-server load-balance method least-outstanding ignore-preferred-server batch-size** *size*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**radius-server load-balance method least-outstanding batch-size** *size* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server load-balance method least-outstanding batch-size 500`	Configures the radius load-balancing options by picking the server with the least-outstanding transactions. This load-balancing method uses the batch-size for the selection of the server. The size ranges from 1 to 1500. If no value is specified, the default is 25.
Step 3	**radius-server load-balance method least-outstanding ignore-preferred-server batch-size** *size* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server load-balance method least-outstanding ignore-preferred-server batch-size 500`	Configures the radius load-balancing options by disabling the preferred server for this Server Group. This load-balancing method uses the batch-size for the selection of the server. The size ranges from 1 to 1500. If no value is specified, the default is 25.
Step 4	**commit**	

Configuring Load Balancing for RADIUS Server: An example

```
configure
radius-server load-balance method least-outstanding batch-size 25
radius-server load-balance method least-outstanding ignore-preferred-server batch-size 45
!
end
```

Configuring Load Balancing for a Named RADIUS Server Group

Perform this task to activate the load balancing function for a named RADIUS server group. As an example, in this configuration the preferred server is set to be ignored.

SUMMARY STEPS

1. **configure**
2. **aaa group server radius** *server_group_name* **load-balance method least-outstanding batch-size** *size*
3. **aaa group server radius** *server_group_name* **load-balance method least-outstanding ignore-preferred-server batch-size** *size*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**aaa group server radius** *server_group_name* **load-balance method least-outstanding batch-size** *size* **Example:** `RP/0/RSP0/CPU0:router(config)# aaa group server radius sg1 load-balance method least-outstanding batch-size 500`	Configures the radius load-balancing options by picking the server with the least-outstanding transactions. This load-balancing method uses the batch-size for the selection of the server. The size ranges from 1 to 1500. If no value is specified, the default is 25.
Step 3	**aaa group server radius** *server_group_name* **load-balance method least-outstanding ignore-preferred-server batch-size** *size* **Example:** `RP/0/RSP0/CPU0:router(config)# aaa group server radius sg1 load-balance method least-outstanding ignore-preferred-server batch-size 500`	Configures the radius load-balancing options by disabling the preferred server for this Server Group. This load-balancing method uses the batch-size for the selection of the server. The size ranges from 1 to 1500. If no value is specified, the default is 25.
Step 4	**commit**	

Throttling of RADIUS Records

The Throttling of AAA (RADIUS) records is a mechanism to avoid RADIUS congestion and instability. This function is useful in situations when there is insufficient bandwidth to accommodate a sudden burst of AAA requests generated by the BNG for the RADIUS server.

While configuring throttling, a threshold rate, which corresponds to the maximum number of outstanding requests, is defined. It is possible to configure independent throttling rates for access (authentication and authorization) and accounting requests. After a threshold value is reached for a server, no further requests of that type are sent to the server. However, for the pending requests, a retransmit timer is started, and if the outstanding request count (which is checked after every timer expiry), is less than the threshold, then the request is sent out.

As a session may timeout due to throttle on the access requests, a limit is set for the number of retransmit attempts. After this limit is reached, further access requests are dropped. Throttled accounting requests, however, are processed through the server-group failover process.

The throttling feature can be configured globally, or for a server-group. However, the general rule of configuration preference is that the server-group configuration overrides global configuration, if any.

The syntax for the throttling CLI command is:

```
radius-server throttle {[accounting THRESHOLD] [access THRESHOLD [access-timeout
NUMBER_OF-TIMEOUTS]]}
```

where:

- accounting THRESHOLD—Specifies the threshold for accounting requests. The range is from 0 to 65536. The default is 0, and indicates that throttling is disabled for accounting requests.
- access THRESHOLD—Specifies the threshold for access requests. The range is from 0 to 65536. The default is 0, and indicates that throttling is disabled for accounting requests.
- access-timeout NUMBER_OF-TIMEOUTS—Specifies the number of consecutive timeouts that must occur on the router, after which access-requests are dropped. The range of is from 0 to 10. The default is 3.

Note By default, the throttling feature is disabled on BNG.

For activating throttling globally, see Configuring RADIUS Throttling Globally, on page 38.

For activating throttling on a server group, see Configuring RADIUS Throttling on a Server Group, on page 39.

Configuring RADIUS Throttling Globally

Perform this task to activate RADIUS throttling globally.

SUMMARY STEPS

1. **configure**
2. **radius-server throttle access** *threshold_value*
3. **radius-server throttle access** *threshold_value* **access-timeout** *value*
4. **radius-server throttle access** *threshold_value* **access-timeout** *value* **accounting** *threshold_value*
5. **radius-server throttle accounting** *threshold_value* **access** *value* **access-timeout** *value*
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**radius-server throttle access** *threshold_value* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server throttle access 10`	Controls the number of access requests sent to a RADIUS server. The threshold value denotes the number of outstanding access requests after which throttling should be performed. The range is from 0 to 65535, and the preferred value is 100.

	Command or Action	Purpose
Step 3	**radius-server throttle access** *threshold_value* **access-timeout** *value* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server throttle access 10 access-timeout 5`	Specifies the number of timeouts, after which a throttled access request is dropped. The value denotes the number of timeouts for a transaction. The range is from 1 to 10, and the default is 3.
Step 4	**radius-server throttle access** *threshold_value* **access-timeout** *value* **accounting** *threshold_value* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server throttle access 10 access-timeout 5 accounting 10`	Controls the number of access timeout requests sent to a RADIUS server. The threshold value denotes the number of outstanding accounting transactions after which throttling should be performed. The range is from 0 to 65535, and the preferred value is 100.
Step 5	**radius-server throttle accounting** *threshold_value* **access** *value* **access-timeout** *value* **Example:** `RP/0/RSP0/CPU0:router(config)# radius-server throttle accounting 56 access 10 access-timeout 5`	Controls the number of accounting requests sent to a RADIUS server. The threshold value denotes the number of outstanding accounting transactions after which throttling should be performed. The value ranges between 0 to 65535 and the preferred value is 100.
Step 6	**commit**	

Configuring RADIUS Throttling Globally: An example

```
configure
radius-server throttle access 10 access-timeout 5 accounting 10
!
end
```

Configuring RADIUS Throttling on a Server Group

Perform this task to activate RADIUS throttling on a server group.

SUMMARY STEPS

1. **configure**
2. **aaa group server radius** *server_group_name*
3. **server** *hostname* **acct-port** *acct_port_value* **auth-port** *auth_port_value*
4. **throttle access** *threshold_value* **access-timeout** *value* **accounting** *threshold_value*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**aaa group server radius** *server_group_name* **Example:** `RP/0/RSP0/CPU0:router(config)# aaa group server radius SG1`	Configures the AAA (RADIUS) server-group definition.
Step 3	**server** *hostname* **acct-port** *acct_port_value* **auth-port** *auth_port_value* **Example:** `RP/0/RSP0/CPU0:router(config-sg-radius)# server 99.1.1.10 auth-port 1812 acct-port 1813`	Configures a RADIUS server accounting or authentication port with either the IP address or hostname (as specified). The accounting port number and the authentication port number ranges from 0 to 65535.
Step 4	**throttle access** *threshold_value* **access-timeout** *value* **accounting** *threshold_value* **Example:** `RP/0/RSP0/CPU0:router(config-sg-radius)# radius-server throttle access 10 access-timeout 5 accounting 10`	Configures the RADIUS throttling options to control the number of access and accounting requests sent to a RADIUS server. The threshold value denotes the number of outstanding access requests or accounting transactions after which throttling should be performed. The range is from 0 to 65535, and for both access and accounting requests the preferred value is 100.
Step 5	**commit**	

Configuring RADIUS Throttling on a Server Group: An example

```
configure
aaa group server radius SG1
server 99.1.1.10 auth-port 1812 acct-port 1813
radius-server throttle access 10 access-timeout 5 accounting 10
!
end
```

RADIUS Change of Authorization (CoA) Overview

The 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-logon — When a user logs into a network, an external web portal that supports CoA sends an account-logon request to BNG with the user's credentials (username and password). Account-logon on BNG then attempts to authenticate the user through RADIUS with those credentials.
- account-logoff— BNG processes the account-logoff request as a disconnect event for the subscriber and terminates the session.

> **Note** The RADIUS CoA server does not differentiate between originators of the disconnect event. Hence, when the BNG receives an account-logoff request from the RADIUS CoA server, for both a user-initiated and an administrator-initiated request, the Acct-Terminate-Cause to be sent to the RADIUS server is always set as Admin-Reset.

- 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 347.

> **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.

Multi-Action Change of Authorization

BNG supports multi-action Change of Authorization (CoA) wherein service providers can activate and deactivate multiple services using a single CoA request. Multi-action CoA is supported for **Service-Logon** and **Service-Logoff** CoA commands. The Service-Logon command can contain one or more **Service-Activate** attributes, and optionally **Service-Deactivate** attributes, for multi-action CoA to specify service(s) to be activated or deactivated. Similarly, the **Service-Logoff** command can contain one or more **Service-Deactivate** attributes, and optionally **Service-Activate** attributes, for multi-action CoA to specify service(s) to be deactivated or activated.

MA-CoA supports up to a maximum of 10 service activations or deactivations per MA-CoA request, however, it is recommended to issue six activations or deactivations per MA-CoA request.

During the multi-action CoA request, if any of the COA requests fail to activate or deactivate, then any of the services which have been activated or deactivated as part of that CoA request is rolled back to its previous state. The session restores back to the its pre-MA-CoA state upon failure to activation or deactivation.

A rollback-failure event, exception, can be configured to specify what action to be taken when a service rollback fails following a failed MA-CoA request (that is, a case of a double-failure condition). The default action to be taken when the rollback fails is to preserve the session, however, you can configure to terminate the session.

The following example details on the rollback failure exception.

```
policy-map type control subscriber PL1
 event session-start match-first
  class type control subscriber class-default do-all
   1 activate dynamic-template pkt-trig1
  !
 !
 event exception match-first
  class type control subscriber coa-rollback-failure do-all
   10 disconnect
  !
 !
!
```

An Example of a Multi-Action Change of Authorization Use Case

The following example lists the sequence of events that occur in the case of a PTA session initiation.

1. PTA session's web traffic redirected to a service portal (HTTP Redirect)
2. The user activates the first level of service through the service portal. A multi-action COA request is initiated in the following sequence.
    1. Deactivate redirection
    2. Activate Turbo Button 1
    3. Activate VoIP with two channels
3. The user activates the second level of service through the service portal. A multi-action COA request is initiated in the following sequence.
    1. Deactivate Turbo Button 1
    2. Activate Turbo Button 2

3. Deactivate VoIP with two channels
4. Activate VoIP with 4 channels

Interworking with Service-Level Accounting

BNG supports Service-Level Accounting, where a service is a collection of features that are activated and deactivated as a group. Service-Level Accounting and MA-CoA features are independent, that is, they can be applied separately. However, MA-CoA accounts for services that are activated or deactivated that have Service-Level Accounting enabled through the dynamic template configuration.

Generating Accounting Records

The following cases describes how the multi-action CoA records are generated for accounting purposes.

MA-CoA ACK Case

- If MA-CoA request contains only service activate commands, then START accounting record for those services are generated after the CoA Ack is sent out.
- If MA-CoA request contains only deactivate services or combination of activate and deactivate services, then for those services START or STOP accounting records are generated after the CoA Ack is sent out.

MA-CoA NAK Case (Rollback scenario)

- If MA-CoA request fails due to presence of invalid command formats or due to internal software failure or due to presence of invalid service names, that are not defined in the box, in such cases the accounting START or STOP messages are not generated upon rollback.
- If MA-CoA request fails due to internal feature programming failure, then the Service-START or Service-STOP accounting records may be generated for the services that were activated or deactivated before the failure. After the failure, the rollback is initiated and appropriate Service-START or Service-STOP records are generated for these services.

High Availability for MA-CoA

If an high availability event other than a line card online insertion and removal (LC-OIR), such as a process restart or an RP failover occurs while an MA-CoA request is being processed, then the affected session is restored to its pre-MA-CoA state. The policy plane does not make an attempt to automatically recover the MA-CoA message or to resume processing. Instead, the CoA Client times out and re-sends the MA-CoA request to the BNG router.

An Example with Verification Commands

The following example shows the profile of a subscriber with existing services, modified with a MA-CoA request, and the subscriber profile with the changed services invoked by the MA-CoA request.

Multi-Action Change of Authorization - Verification Commands

Session with an Existing Service --[1]

```
show subsscriber session all detail internal
```

```
Interface:                 Bundle-Ether1.1.ip1
Circuit ID:                Unknown
Remote ID:                 Unknown
Type:                      IP: DHCP-trigger
IPv4 State:                Up, Wed Jul  9 14:25:40 2014
IPv4 Address:              12.1.0.2, VRF: default
IPv4 Up helpers:           0x00000040 {IPSUB}
IPv4 Up requestors:        0x00000040 {IPSUB}
Mac Address:               0000.0c00.0001
Account-Session Id:        00000001
Nas-Port:                  Unknown
User name:                 0000.0c00.0001
Outer VLAN ID:             10
Subscriber Label:          0x00000040
Created:                   Wed Jul  9 14:25:37 2014
State:                     Activated
Authentication:            unauthenticated
Authorization:             authorized
Ifhandle:                  0x020001a0
Session History ID:        1
Access-interface:          Bundle-Ether1.1
Policy Executed:

  event Session-Start match-first [at Wed Jul  9 14:25:37 2014]
    class type control subscriber ISN_CM do-all [Succeeded]
      1 activate dynamic-template ISN_TEMPLATE_1 [cerr: No error][aaa: Success]
      2 authorize aaa list default [cerr: No error][aaa: Success]
      1001 activate dynamic-template svcQoSAcct2 [cerr: No error][aaa: Success]
      1002 activate dynamic-template svcQoSAcct3 [cerr: No error][aaa: Success]
Session Accounting: disabled
Last COA request received: unavailable
User Profile received from AAA:
 Attribute List: 0x1000eb24
 1:  ipv4-mtu        len=  4  value= 1500(5dc)
Services:
  Name        : ISN_TEMPLATE_1
  Service-ID  : 0x4000002
  Type        : Template
  Status      : Applied
-------------------------
  Name        : svcQoSAcct1
  Service-ID  : 0x400000a
  Type        : Multi Template
  Status      : Applied
-------------------------
  Name        : svcQoSAcct2
  Service-ID  : 0x400000b
  Type        : Template
  Status      : Applied
-------------------------
  Name        : svcQoSAcct3
  Service-ID  : 0x400000c
  Type        : Template
  Status      : Applied
-------------------------
[Event History]
   Jul  9 14:29:41.056 IPv4 Start
   Jul  9 14:29:44.384 SUBDB produce done
   Jul  9 14:29:44.384 IPv4 Up

RP/0/RSP1/CPU0:BNG#show subscriber database association
```

```
Location 0/RSP1/CPU0

Bundle-Ether1.1.ip1, subscriber label 0x40
  Name                            Template Type
  --------                        -------------
  U00000040                       User profile
  svcQoSAcct3                     Service
  svcQoSAcct2                     Service
  svcQoSAcct1                     Service
  ISN_TEMPLATE_1                  IP subscriber
```

MA-CoA Request Initiated From RADIUS Client --
[2]

```
exec  /bin/echo
"Cisco-AVPair='subscriber:sd=svcQoSAcct1',Cisco-AVPair='subscriber:sd=svcQoSAcct2',
Cisco-AVPair='subscriber:sd=svcQoSAcct3',Cisco-AVPair='subscriber:sa=qosin_coa',
Cisco-AVPair='subscriber:sa=qosout_coa',Acct-Session-Id=00000001" | /usr/local/bin/radclient
 -r 1 -x 5.11.17.31:1700 coa coa


RP/0/RSP1/CPU0:BNG#show subscriber manager statistics AAA COA location 0/rsp1/cpu0


[ CHANGE OF AUTHORIZATION STATISTICS ]

CoA Requests:
                 Type     Received        Acked        NAKed
                 ====     ========        =====        =====
        Account Logon            0            0            0
       Account Logoff            0            0            0
       Account Update            0            0            0
           Disconnect            0            0            0
 Single Service Logon            0            0            0
Single Service Logoff            0            0            0
Single Service Modify            0            0            0
     Multiple Service            1            1            0

Errors:
  None

RP/0/RSP1/CPU0:BNG#show subscriber session all detail internal

Interface:                Bundle-Ether1.1.ip1
Circuit ID:               Unknown
Remote ID:                Unknown
Type:                     IP: DHCP-trigger
IPv4 State:               Up, Wed Jul  9 14:25:40 2014
IPv4 Address:             12.1.0.2, VRF: default
IPv4 Up helpers:          0x00000040 {IPSUB}
IPv4 Up requestors:       0x00000040 {IPSUB}
Mac Address:              0000.0c00.0001
Account-Session Id:       00000001
Nas-Port:                 Unknown
User name:                0000.0c00.0001
Outer VLAN ID:            10
Subscriber Label:         0x00000040
Created:                  Wed Jul  9 14:25:37 2014
State:                    Activated
Authentication:           unauthenticated
Authorization:            authorized
```

```
Ifhandle:                 0x020001a0
Session History ID:       1
Access-interface:         Bundle-Ether1.1
Policy Executed:

  event Session-Start match-first [at Wed Jul  9 14:25:37 2014]
    class type control subscriber ISN_CM do-all [Succeeded]
      1 activate dynamic-template ISN_TEMPLATE_1 [cerr: No error][aaa: Success]
      2 authorize aaa list default [cerr: No error][aaa: Success]
      1001 activate dynamic-template svcQoSAcct2 [cerr: No error][aaa: Success]
      1002 activate dynamic-template svcQoSAcct3 [cerr: No error][aaa: Success]
Session Accounting: disabled
Last COA request: Wed Jul  9 14:27:37 2014
COA Request  Attribute List: 0x1000f0c4
 1:  sd              len= 11  value= svcQoSAcct1
 2:  command         len= 18  value= deactivate-service
 3:  service-info    len= 11  value= svcQoSAcct1
 4:  service-name    len= 11  value= svcQoSAcct1
 5:  sd              len= 11  value= svcQoSAcct2
 6:  command         len= 18  value= deactivate-service
 7:  service-info    len= 11  value= svcQoSAcct2
 8:  service-name    len= 11  value= svcQoSAcct2
 9:  sd              len= 11  value= svcQoSAcct3
 10:  command         len= 18  value= deactivate-service
 11:  service-info    len= 11  value= svcQoSAcct3
 12:  service-name    len= 11  value= svcQoSAcct3
 13:  sa              len=  9  value= qosin_coa
 14:  command         len= 16  value= activate-service
 15:  service-info    len=  9  value= qosin_coa
 16:  service-name    len=  9  value= qosin_coa
 17:  sa              len= 10  value= qosout_coa
 18:  command         len= 16  value= activate-service
 19:  service-info    len= 10  value= qosout_coa
 20:  service-name    len= 10  value= qosout_coa
Last COA response: Result ACK
COA Response  Attribute List: 0x1000f4e4
 1:  sd              len= 11  value= svcQoSAcct1
 2:  sd              len= 11  value= svcQoSAcct2
 3:  sd              len= 11  value= svcQoSAcct3
 4:  sa              len=  9  value= qosin_coa
 5:  sa              len= 10  value= qosout_coa
User Profile received from AAA:
 Attribute List: 0x1000f6f4
 1:  ipv4-mtu        len=  4  value= 1500(5dc)
Services:
  Name        : ISN_TEMPLATE_1
  Service-ID  : 0x4000002
  Type        : Template
  Status      : Applied
-------------------------
  Name        : qosin_coa
  Service-ID  : 0x4000006
  Type        : Multi Template
  Status      : Applied
-------------------------
  Name        : qosout_coa
  Service-ID  : 0x4000008
  Type        : Multi Template
  Status      : Applied
-------------------------
[Event History]
   Jul  9 14:29:41.056 IPv4 Start
   Jul  9 14:29:44.384 IPv4 Up
```

```
   Jul  9 14:31:41.504 CoA request
   Jul  9 14:31:41.632 SUBDB produce done [many]
```

Changed Subscriber Profile after the MA-CoA Request is Processed from RADIUS
--[3]

```
RP/0/RSP1/CPU0:BNG#show subscriber database association


Location 0/RSP1/CPU0

Bundle-Ether1.1.ip1, subscriber label 0x40
  Name                            Template Type
  --------                        -------------
  U00000040                       User profile
  qosout_coa                      Service
  qosin_coa                       Service
  ISN_TEMPLATE_1                  IP subscriber
```

In the above example, the subscriber profile existing services are defined by [1], the changes initiated by the MA-CoA request is represented by [2], and the changes that are impacted by the MA-CoA request is shown in [3].

Restrictions in Multi-Action Change of Authorization

Multi-Action Change of Authorization is subjected to the following restrictions:

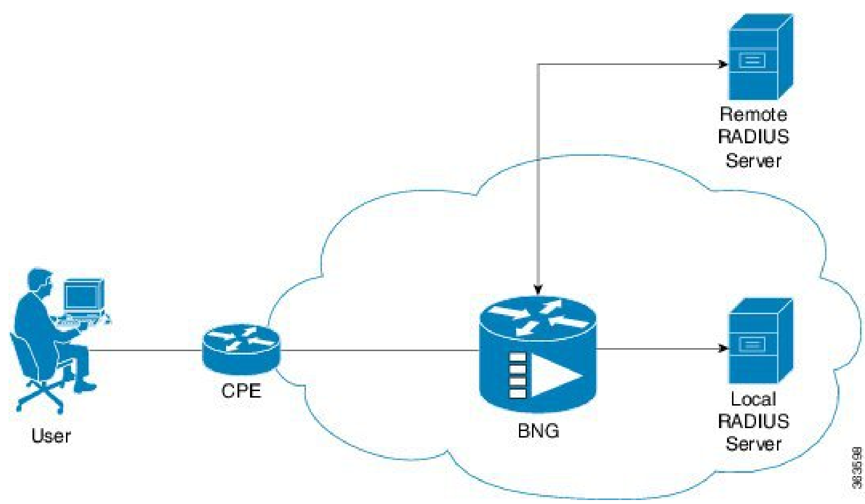
- **Service-Activate and Service-Deactivate commands only**: Only the Service-Activate and Service-Deactivate commands are supported in the MA-CoA requests. If a MA-CoA request containing account-logon, account-logoff, account-update, session-query, or disconnect-request commands is received, the request is rejected.
- Cisco VSAs of format **"subscriber:command= activate-service"** and **"subscriber:service-name=Svc1**" are not supported in MA-COA. If requests containing these VSA formats are received, a NAK is sent. Only formats of the **"subscriber:sa/sd=svcname"** type is supported.
- Event **service-logon** and **service-logoff** actions are not supported under policy map for services activated or deactivated through MA-CoA (same as service activation done as part of Access-Accept).
- MA-CoA with QoS Shaper Parameterization is not supported.
- **Dynamic Template services only**: MA-CoA is supported on services that are defined through dynamic templates configured on the router. MA-CoA design does not preclude support for services downloaded through RADIUS server, that is, the service profiles.
- **CoA Account-Update messages must not contain any Service-Activate or Deactivate VSAs**: MA-CoA does not restrict or detect Service-Activate or Service-Deactivate VSAs within the CoA Account-Update messages, however, the support is not available.
- **Bundle Subscribers only**: MA-COA is currently supported on RP-based subscribers (bundle-access interfaces) only.
- **Scale, Performance, Boundary Conditions**: The following are the conditions for MA-CoA:
    - MA-CoA does not to impose any significant limitations on scaling, in terms of the total number of sessions or the number of services applied per session.

- MA-CoA supports up to a maximum of 10 service activations or deactivations per MA-CoA request. If the number of action requests exceeds the limit of 10, a NAK is initiated for the last request received.
- MA-CoA can handle a maximum of 30 CoA messages per second.

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.

Figure 4: User Authentication and Authorization in the Local Network



User Authentication and Authorization in the Local Network feature is used in a case when a user wants to perform a two-level authentication or authorization, first, a remote authentication (or authorization) followed by a local authorization (or authentication).

Note All the debug commands applicable to AAA server are applicable on User Authentication and Authorization in the Local Network feature.

For IPoE subscribers, User Authentication and Authorization in the Local Network is a two-level authorization process as a part of the session-start event. For PTA subscribers, User Authentication and Authorization in the Local Network is a remote server authentication process, followed by a local server authorization process.

Policy Configurations for IPoE Sessions

The following policy configuration explains how the authentication and authorization process occurs in IPoE subscriber sessions. The authentication and authorization processes are performed using two RADIUS servers (one located remotely and the other located locally). At first, the authentication request is routed to the remotely located RADIUS server, which is not in the user's control. Then, to authorize the session, the authorization request is routed to the local RADIUS server, where the subscriber profile for the service provider is maintained.

As a first step in the authorization process, you can configure the authentication process to download the authorization profile from the local RADIUS server. However, when both RADIUS servers have the same authorization profiles, either partially or completely, that part of the authorization profile that is the same is overridden by the one downloaded from the local RADIUS server, and the other part of the authorization profile is merged.

Case 1: Subscriber session created by applying the user profile downloaded from the local RADIUS server.

```
Radius Server1 (located remotely, profile not controlled by the operator

              0000.0000.0001  Cleartext-Password := "shootme"
             Fall-Through = no

Radius Server2 (located locally, profile controlled by the operator)

              0000.0000.0001  Cleartext-Password := "shootme"
             Class = "IPSUB",
             Cisco-avpair += "ip:sub-qos-policy-in=12MUp",
             Cisco-avpair += "ip:sub-qos-policy-out=12MDown",
             Fall-Through = no
```

Case 2: Subscriber session created by applying the user profile downloaded from the remote RADIUS server, and in this case, the policy attribute values are overridden by the local RADIUS server profile.

```
Radius Server1 (located remotely, profile not controlled by the operator)

             0000.0000.0001  Cleartext-Password := "shootme"
             Cisco-avpair += "ip:sub-qos-policy-in=6MUp",
             Cisco-avpair += "ip:sub-qos-policy-out=6MDown",
             Fall-Through = no

Radius Server2 (located locally, profile controlled by the operator)

              0000.0000.0001  Cleartext-Password := "shootme"
             Class = "IPSUB",
             Cisco-avpair += "ip:sub-qos-policy-in=12MUp",
             Cisco-avpair += "ip:sub-qos-policy-out=12MDown",
             Fall-Through = no
```

Profile Created by the Attribute Merging of both the Local and Remote Server Profiles

```
RP/0/RSP0/CPU0:BNG#sh run aaa
radius-server host 10.105.236.46 auth-port 1812 acct-port 1813
key 7 111B1801464058
!
radius-server host 10.105.236.237 auth-port 1812 acct-port 1813
key 7 095E4F0D485744
```

```
!
aaa group server radius local_server
server 10.105.236.237 auth-port 1812 acct-port 1813
!
aaa group server radius remote_server
server 10.105.236.46 auth-port 1812 acct-port 1813
!
aaa accounting subscriber acct_meth broadcast group local_server group remote_server
aaa authorization subscriber local_server group local_server
aaa authorization subscriber remote_server group remote_server

RP/0/RSP0/CPU0:BNG#

RP/0/RSP0/CPU0:BNG#sh run policy-map type control subscriber ISN_CNTRL_1
policy-map type control subscriber ISN_CNTRL_1
event session-start match-all
  class type control subscriber ISN_CM do-all
   10 activate dynamic-template ISN_TEMPLATE_1
   11 authorize aaa list remote_server identifier source-address-mac password shootme
   12 authorize aaa list local_server identifier source-address-mac password shootme
  !
!
end-policy-map
!

RP/0/RSP0/CPU0:BNG#
Remote User Profile
0000.0c00.0001 Cleartext-Password := "shootme"
    cisco-avpair += "subscriber:accounting-list=acct_meth", -- [(A) Same attribute on both
 profile]
    Session-Timeout += 1000, --------------------------------- [(B) Attribute defined in
remote profile only]
    Acct-Interim-Interval = 3600 ----------------------------- [(C) Same attribute on both
 profiles with diff value}


Local User profile
0000.0c00.0001 Cleartext-Password := "shootme"
    cisco-avpair += "subscriber:accounting-list=acct_meth", -- [(A) Same attribute on both
 profile]
    cisco-avpair += "sub-qos-policy-in=12MUp",---------------- [(D) Attribute defined in
local profile only]
    cisco-avpair += "sub-qos-policy-out=12MDown",------------- [(E) Attribute defined in
local profile only]
    cisco-avpair += "ipv4:inacl=innet", ---------------------- [(F) Attribute defined in
local profile only]
    cisco-avpair += "ipv4:outacl=outnet", -------------------- [(G) Attribute defined in
local profile only]
    Acct-Interim-Interval = 3000 ----------------------------- [(H) Same attributes on both
 profiles with diff value]


RP/0/RSP0/CPU0:BNG#sh subscriber session all detail internal
Interface:                Bundle-Ether1.1.ip22
Circuit ID:               Unknown
Remote ID:                Unknown
Type:                     IP: DHCP-trigger
IPv4 State:               Up, Wed Jun 18 16:56:25 2014
IPv4 Address:             12.16.0.24, VRF: default
IPv4 Up helpers:          0x00000040 {IPSUB}
IPv4 Up requestors:       0x00000040 {IPSUB}
Mac Address:              0000.0c00.0001
Account-Session Id:       000000bb
Nas-Port:                 Unknown
```

```
User name:                0000.0c00.0001
Outer VLAN ID:            10
Subscriber Label:         0x00000075
Created:                  Wed Jun 18 16:56:15 2014
State:                    Activated
Authentication:           unauthenticated
Authorization:            authorized
Ifhandle:                 0x000012a0
Session History ID:       11
Access-interface:         Bundle-Ether1.1
Policy Executed:


  event Session-Start match-all [at Wed Jun 18 16:56:15 2014]
    class type control subscriber ISN_CM do-all [Succeeded]
      10 activate dynamic-template ISN_TEMPLATE_1 [cerr: No error][aaa: Success]
      11 authorize aaa list remote_server [cerr: No error][aaa: Success]
      12 authorize aaa list local_server [cerr: No error][aaa: Success]
Session Accounting:
  Acct-Session-Id:          000000bb
  Method-list:              acct_meth
  Accounting started:       Wed Jun 18 16:56:25 2014
  Interim accounting:       On, interval 50 mins
    Last successful update: Never
    Next update in:         00:46:48 (dhms)
    Last update sent:       Never
    Updates sent:           0
    Updates accepted:       0
    Updates rejected:       0
    Update send failures:   0
Last COA request received: unavailable
User Profile received from AAA:
Attribute List: 0x1000e764
1:  session-timeout len=  4  value= 1000(3e8)  ------- [(B) Attribute value fetched from
the remote profile]
2:  accounting-list len=  9  value= acct_meth  ------- [(A) Attribute common to both the
profiles]
3:  sub-qos-policy-in len=  5  value= 12MUp    ------- [(D) Attribute defined in the local
 profile]
4:  sub-qos-policy-out len=  7  value= 12MDown ------- [(E) Attribute defined in the local
 profile]
5:  inacl           len=  5  value= innet ------------ [(F) Attribute defined in the local
 profile]
6:  outacl          len=  6  value= outnet ----------- [(G) Attribute defined in the local
 profile]
7:  acct-interval   len=  4  value= 3000(bb8) -------- [(I) Attribute value fetched from
the local profile]
Services:
  Name        : ISN_TEMPLATE_1
  Service-ID  : 0x4000002
  Type        : Template
  Status      : Applied
-------------------------
```

In the above example, the server profile attributes are defined in both the Local RADIUS and the Remote RADIUS servers. Attributes (A), (B), and (C) are defined in remote RADIUS server profile, and attributes (A), (D), (E), (F), (G), and (H) are defined in the local RADIUS server profile. The subscriber session created by applying the user profile downloaded from the local RADIUS server contains attributes (B), (A), (D), (E), (F), (G), and (I), where the attribute (B) is fetched from the remote RADIUS server profile; the attribute (A) is common to both the RADIUS server profiles; the attributes (D), (E), (F), and (G) are the attributes fetched from the local RADIUS server profile; and attribute (I) is common to both the profiles, however, the attribute value differs on both the profiles. In this case, the value of the attribute (I) is fetched from the local RADIUS server profile.

Policy Configurations for PTA Sessions

The following policy configuration explains how the authentication and authorization processes occur in PTA subscriber sessions. In the case of PTA subscriber sessions, the authentication and authorization processes consists of two steps:

1. Domain Authorization on LAC: Achieved through the local RADIUS server where the domain authorization occurs.

```
policy-map type control subscriber vpdn_ipv4_pmap
event session-start match-first
 class type control subscriber vpdn_ipv4_cmap do-until-failure
  ! activate dynamic-template vpdn_v4
 !
event session-activate match-first
 class type control subscriber vpdn_ipv4_cmap do-until-failure
  10 authorize aaa list vpdn-author-list format vpdn_domain password cisco
  20 authenticate aaa list vpdn-authen-list
 !
```

2. User Authentication before forwarding on LAC: Achieved using two RADIUS servers: one local RADIUS server for domain authorization and another remote RADIUS server for user authentication.

```
radius-server vsa attribute ignore unknown

radius-server host 5.8.23.156 auth-port 1812 acct-port 1813
key 7 02050D480809
!
radius-server host 5.8.23.160 auth-port 1812 acct-port 1813
key 7 030752180500
!
radius-server key 7 0214055F5A545C
aaa attribute format vpdn_domain
username-strip prefix-delimiter @
!
aaa accounting network default start-stop group radius
aaa group server radius vpdn-authen
server 5.8.23.160 auth-port 1812 acct-port 1813
!
aaa group server radius vpdn-author
server 5.8.23.156 auth-port 1812 acct-port 1813
!
aaa accounting subscriber default group radius
aaa authorization subscriber vpdn-author-list group vpdn-author
aaa authentication subscriber vpdn-authen-list group vpdn-authen
!
```

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.

Service accounting starts when a subscriber session comes up with a service enabled on it. This can happen through a dynamic template applied through a control policy, through access-accept (AA) messages when the session is authorized, or through a change of authorization (CoA), when a new service is applied on a subscriber session. Service accounting stops either when the session is terminated, or a service is removed from the session through CoA, or some other event that deactivates the service. Start records have no counters; interim and stop records with QoS counters are generated when service accounting is enabled for QoS. Interim accounting records can be generated, in between start and stop accounting, as an option with a pre-defined periodic interval. When the interim period is zero, interim accounting records are not created. Different interim intervals are based on every service for each session. Service accounting is enabled on each template, based on the configuration.

Service Accounting is supported on bundle subscriber interfaces as well as line card subscriber interfaces.

Note The policy-map associated to a dynamic template can be edited to change the service parameters. However, this does not update the accounting records. Therefore, to generate all the accounting records accurately, it is recommended that a new service with all the required service parameters be created and associated to the new service, through a CoA.

For service accounting, statistics for ingress and egress QoS policies, which are applied under each service for a given subscriber, may need to be reported as part of the accounting interim and stop records. For each service, these QoS counters can be reported as part of the accounting records:

- BytesIn — Aggregate of bytes matching all classes of the ingress QoS policy for the service minus the policer drops.
- PacketsIn — Aggregate of packets matching all classes of the ingress QoS policy for the service minus the policer drops.
- BytesOut — Aggregate of bytes matching all classes of the egress QoS policy for the service minus the queuing drops.
- PacketsOut — Aggregate of packets matching all classes of the egress QoS policy for the service minus the queuing drops

Dynamic template features that support accounting statistic collection and require that their statistics be reported in the AAA service accounting records can enable accounting statistics on their features using the newly-introduced optional **acct-stats** configuration option. This option is not available for the features that do not support statistic collection. By default, QoS accounting statistics are disabled to optimize performance.

Note The QoS counters for each direction is reported only if a QoS policy is applied for that service in the given direction. For example, if a service does not have an ingress policy applied, BytesIn and PacketsIn counters are reported as being 0.

Pre-requisites

- Subscriber accounting, the parent accounting record for service accounting, must be configured to enable the service accounting feature to work.
- The keyword **acct-stats** must be configured in service-policy configuration to enable the service accounting feature to report feature counter information as part of the records.

Restriction

- IPv4 and IPv6 subscriber sessions has a single set of service accounting records. They are merged into one set of bytes_in, bytes_out, packets_in, packets_out counters.
- Service accounting is not supported for static sessions.

Configuring Service Accounting

Perform this task to configure service accounting through the dynamic template:

Before you begin

You must configure subscriber accounting before performing this task. Refer Creating Dynamic Template for IPv4 or IPv6 Subscriber Session, on page 77 for configuring procedure.

SUMMARY STEPS

1. **configure**
2. **aaa accounting service** {*list_name* | **default**} {**broadcast group** {*group_name* | **radius**} | **group** {*group_name* | **radius**} }
3. **aaa service-accounting** [**extended** | **brief**]
4. **dynamic-template**
5. **type service** *dynamic-template-name*
6. **accounting aaa list** {*method_list_name* | **default**} **type service** [**periodic-interval** *time*]
7. {**ipv4** | **ipv6**} **access-group** *access-list-name*
8. **service-policy** {**input** | **output** | **type**} *service-policy_name* [**acct-stats**]
9. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**aaa accounting service** {*list_name* | **default**} {**broadcast group** {*group_name* | **radius**} | **group** {*group_name* | **radius**} } **Example:** `RP/0/RSP0/CPU0:router(config)# aaa accounting service l1 group srGroup1`	Creates an accounting list for service accounting
Step 3	**aaa service-accounting** [**extended** | **brief**] **Example:** `RP/0/RSP0/CPU0:router(config)# aaa service-accounting brief`	(Optional) Sets accounting parameters for service to select the level of subscriber accounting state and to identity attribute reporting in brief or extended form. **Note** The default setting is extended.
Step 4	**dynamic-template** **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template`	Enters the dynamic-template configuration mode.
Step 5	**type service** *dynamic-template-name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type service s1`	Creates a dynamic-template with a user-defined name for a service.
Step 6	**accounting aaa list** {*method_list_name* | **default**} **type service** [**periodic-interval** *time*] **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# accounting aaa list l1 type service periodic-interval 1000`	Configures the service accounting feature.

Step 7	{**ipv4** | **ipv6**} **access-group** *access-list-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv4 access-group ACL1`
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 access-group ACL2`
		Sets IPv4 or IPv6 access list to an interface.

Step 8	**service-policy** {**input** | **output** | **type**} *service-policy_name* [**acct-stats**]
	Example:
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy input QoS1 acct-stats`
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy output QoS2 acct-stats`
		Associates a service-policy to the dynamic template, and enables service accounting feature using **acct-stats** keyword.

	Command or Action	Purpose
Step 9	**commit**	

Configuring Service Accounting: Example

```
configure
 aaa accounting service S1 group SG1
 aaa service-accounting brief
 dynamic-template
  type service s1
   accounting aaa list S1 type service periodic-interval 600
   ipv4 access-group ACL1
   service-policy input QOS1 acct-stats
   service-policy output QOS2 acct-stats
!
!
end
```

Statistics Infrastructure

The accounting counters are maintained by the service accounting statistics IDs (statsD) infrastructure. Service accounting interacts with the statistics infrastructure in this manner:

- Each feature has a statistics collector process that is responsible for returning statistics counters for that feature.
- A single collector can handle counters for multiple features.
- An accounting process, the service accounting management agent, uses the access library to register for notifications and request statistics, and pushes to a radius server.

There is a polling period to pull the data from statsD. To support sub-second accuracy on stop records, the statistics are immediately pulled when the session is terminated, without waiting for any polling method to get accurate data. The same method is followed by session accounting and service accounting. Sub-second accuracy is not supported for data reported in interim records, because no data is pulled while sending interim accounting records.

Configuring Statistics IDs (statsD)

The statsD is configured to poll feature statistics by default every 900 seconds (that is, every 15 minutes). Perform this task to change the default figure to either increase or decrease the polling interval.

SUMMARY STEPS

1. **configure**
2. **statistics period service-accounting** {*period* | **disable**}
3. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**statistics period service-accounting** {*period* | **disable**} **Example:** `RP/0/RSP0/CPU0:router(config)# statistics period service-accounting 1800`	Sets collection period for statistics collectors for the service accounting feature.
Step 3	**commit**	

Configuring Service Accounting: Example

```
configure
 statistics period service-accounting 1800
end
```

Understanding Per-VRF AAA Function

The Per VRF AAA function allows authentication, authorization, and accounting (AAA) on the basis of virtual routing and forwarding (VRF) instances. This feature permits the Provider Edge (PE) or Virtual Home Gateway (VHG) to communicate directly with the customer's RADIUS server, (which is associated with the customer's Virtual Private Network (VPN)), without having to go through a RADIUS proxy.

ISPs must be able to define operational parameters such as AAA server groups, method lists, system accounting, and protocol-specific parameters, and associate those parameters to a particular VRF instance.

The Per VRF AAA feature is supported with VRF extensions to server-group, RADIUS, and system accounting commands. The list of servers in server groups is extended to include definitions of private servers, in addition to references to the hosts in the global configuration. This allows simultaneous access to both customer servers and global service provider servers. The syntax for the command used to configure per-vrf AAA globally is:

```
radius source-interface subinterface-name [vrf vrf-name]
```

RADIUS Double-Dip Feature

BNG supports the RADIUS double-dip feature, where BNG sends the first authentication or authorization request to a service provider's RADIUS server, which in turn responds with the correct VRF associated with the subscriber session. Subsequently, the BNG redirects the original request, and sends it as a second request, to the correct RADIUS server that is associated with the designated VRF.

RADIUS over IPv6

From Cisco IOS XR Software Release 5.3.1 and later, RADIUS over IPv6 is supported in BNG, thereby allowing IPv6 address also for various RADIUS configurations and CoA client configurations.

These commands are extended to support IPv6 address:

- **radius-server host** (global configuration mode)
- **radius server** (radius server group configuration mode)
- **radius server-private** (radius server group configuration mode)
- **aaa server radius dynamic-author client** (global configuration mode)

For details on configuring RADIUS server group and settings, see Configuring RADIUS Server Group, on page 15 and Configuring RADIUS Server Settings, on page 30.

Additional References

These sections provide references related to implementing RADIUS.

RFCs

Standard/RFC - AAA	
RFC-2865	Remote Authentication Dial In User Service (RADIUS)
RFC-2866	RADIUS Accounting
RFC-2867	RADIUS Accounting Modifications for Tunnel Protocol Support
RFC-2868	RADIUS Attributes for Tunnel Protocol Support
RFC-2869	RADIUS Extensions
RFC-3575	IANA Considerations for RADIUS
RFC-4679	DSL Forum Vendor-Specific RADIUS Attributes
RFC-5176	Dynamic Authorization Extensions to RADIUS

MIBs

MIBs	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 4

Activating Control Policy

A control policy enables the service provider to define certain actions that are performed during various subscriber life-cycle events. This chapter provides information about activating control policy on the BNG router. A control policy is defined using a policy-map. The policy-map contains a set of events - events during which certain actions are performed. The condition for performing an action is defined in a class-map. After a class-map is created, it is included in the policy-map. The policy-map is then activated on the router interface for the policy to take effect. One of the actions that can be performed by the policy map is activating dynamic template. The dynamic template is a container used to group a set of configuration items to be applied to a group of subscribers. This chapter covers the following topics:

Control Policy Overview

A control policy enables the service provider to define actions that must be performed during various subscriber lifecycle events, such as creation of a session, connectivity loss, and so on. For the complete list of events, see Control Policy Events, on page 64.

Different actions can be executed for different subscribers based on various match criteria. Some actions that can be specified in the control policy are:

- Authenticating or authorizing a subscriber by an external AAA server
- Starting subscriber accounting
- Activating specific configurations on the subscriber using dynamic templates

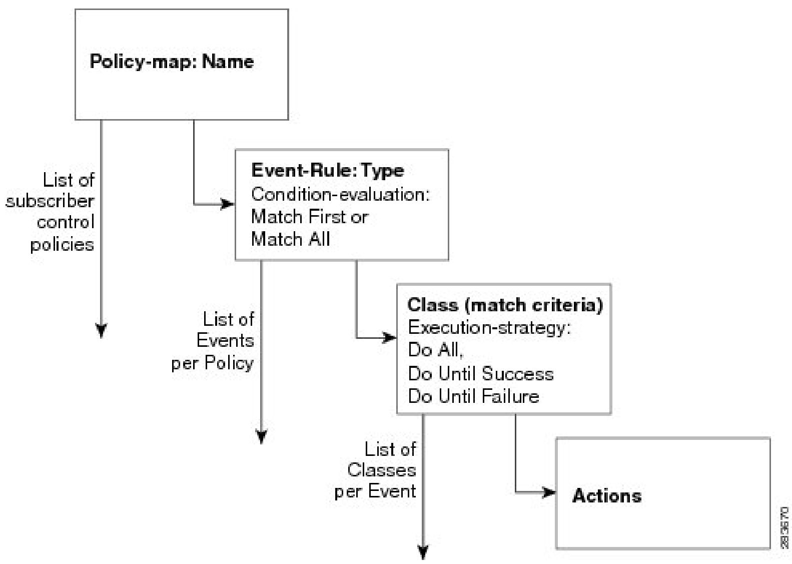
A control policy is deployed using policy-map and class-map. Each policy-map contains a list of events that the service provider considers applicable to the subscriber lifecycle. The policy-map also defines the actions that will be performed during these events. However, these actions are performed only when certain conditions are met. These conditions are called match criteria. The match criteria are defined in class-maps, which is included within the policy-map. It is possible to have different match criteria for different subscribers.

For example, a control policy can be created to start the "subscriber authentication" action, when a "session start" event occurs, for a specific "MAC address" match criteria. After this control policy is deployed, when the device having the specified MAC address starts a new session, BNG initiates the subscriber authentication process.

The actions defined in the policy-map are executed by action handlers. For more information about supported action handlers, see Action Handlers, on page 355.

The following figure shows the structure of control policy. It illustrates that for each policy there can be multiple events; for each event, there can be multiple classes; and for each class, there can be multiple actions. As a result, a single policy map can be used to trigger multiple actions, when a match is found for a single or several criteria, during one or many events.

Figure 5: Control Policy



The following sample configuration shows the control policy structure:

```
policy-map type control subscriber policy-map-name
   event <event-type> [match-all|match-first]
      class type control subscriber <class-map-name>
           <seq#> <action-type> <action_options>
```

Note From Cisco IOS XR Software Release 5.2.2 and later, you can edit the class associated with the subscriber policy even while the sessions are active. Prior to this, new class map actions were not editable if the sessions were up, and any such dynamic policy-map changes resulted in clearing off the subscriber sessions.

Creating Class-Map

The class-map is used to define traffic class. The traffic is classified based on match criteria defined in the class-map. The parameter for match criteria can be protocol, MAC address, input interface, access group, and so on.

If more than one match criteria is listed in a single class-map, then instructions must be included defining how the match criteria are to be evaluated. The evaluation instruction are of two types:

- Match-any—A positive match is made if the traffic being evaluated matches any one of the specified criteria.
- Match-all—A positive match is made only if the traffic being evaluated matches all specified criteria.

Once a match is made, the traffic is considered as a member of the class.

Each class-map is assigned a name for identification. The class-map name is specified within the policy-map.

For creating a class-map, see Configuring a Class-Map, on page 63.

Configuring a Class-Map

Perform this task to configure a class-map for control policies. As an example, this class-map is created with the evaluation instruction, "match-any". The match criteria is "protocol" with value "PPP". As a result, a positive match is made when the session is uses PPP protocol.

SUMMARY STEPS

1. **configure**
2. **class-map type control subscriber match-any** *class-map-name*
3. **match protocol ppp**
4. **end-class-map**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**class-map type control subscriber match-any** *class-map-name* **Example:** `RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-any clmap1`	Creates a new subscriber control class-map with a user defined name. Enters the class-map mode. Defines the match evaluation instruction to be "match-any".
Step 3	**match protocol ppp** **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# match protocol ppp`	Defines the match-criteria to be PPP protocol. **Note** More than one match statement can be applied per class-map.
Step 4	**end-class-map** **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	Ends the class map configuration.
Step 5	**commit**	

Configuring a Class-Map: An example

```
class-map type control subscriber match-any DHCP_class
match protocol dhcpv4
end-class-map
 !
!
end
```

Creating Policy-Map

The policy-map is used to define the events for which a set of actions are executed when a match, based on the class-map definitions, is made. For more information about the supported BNG events, see Control Policy Events, on page 64.

A policy-map lists a set of events. For each event, a set of class-maps are defined. For each class-map, a series of actions are listed sequentially. After the policy-map is applied on the BNG router interface, when the traffic matches the criteria mentioned in the class-map, the actions are performed.

If more than one class-map is listed in the policy-map, then instruction has to be specified that defines which class-maps should be applied. The evaluation instruction are of two types:

- First-match—Actions are performed only when a match is made for the first class-map.
- Match-all—Actions are performed for all matching classes.

Like with a class-map, each policy-map is assigned a name for identification. The policy-map name is specified when activating the policy-map on the router interface.

For creating a policy-map, see Configuring a Policy-Map, on page 65.

Control Policy Events

Control policy on BNG supports the events listed here. These events need to be defined while creating a policy-map using the task Configuring a Policy-Map, on page 65.

- Session-Start—This event is used by the PPPoE and DHCP access protocols to create a subscriber in the policy plane. The operator may configure the AAA actions and activate dynamic templates, suitable for subscriber.
- Session-Activate—Some access protocols require a two-stage session bring-up; for example, with PPPoE subscribers, the PPPoE Access protocol calls the Session-Start event for first sign of life (FSOL), followed by Session-Activate during PPP negotiation and authentication. The operator configures the AAA actions and activates the dynamic templates as suitable for the subscriber.
- Service-Stop—CoA is responsible for generating this event. The BNG operator configures the activate or deactivate actions, to put the subscriber in a default state when a service is stopped.
- Authentication-No-Response—If configured, this event is triggered when there is no response from the AAA server(s) for an authentication request. This event allows the network access server (NAS) operators to define how the failure should be handled. If the authentication-no-response event is not configured, then the authentication failure result is propagated to the access protocol for default handling.

- Authorization-No-Response—If configured, this event is triggered when there is no response from the AAA server(s) for an authorization request. This event allows the NAS operators to define how the failure should be handled. If the authorization-no-response event is not configured, then the authorization results are propagated to the access protocol for default handling, which causes the client who triggered the authorization to disconnect the subscriber session.
- Authentication-Failure—If configured and if the RADIUS server returns an authentication failure, then the Policy Rule Engine returns an "Authentication-Success" to the client that originated the request, in order to prevent it from disconnecting the subscriber. Furthermore, instead of depending on the client to provide the necessary behavior, the actions within the configured Authentication-Failure event are applied on the subscriber.
- Authorization-Failure—The authorization failure event indicates a RADIUS server rejection for the access request. If configured, the service provider overrides the default handling of the failure from the client.
- Timed-Policy-Expiry—If configured, this event is triggered as a result of a policy set-timer action that is configured and set on a subscriber session. This event allows NAS operators to define a timer for a number of possible scenarios. The set timer indicates that certain subscriber state changes have taken place. If sessions are not in the desired state, the NAS operators can disconnect or terminate the session through a configured disconnect action, or impose a different user policy.
- Account-Logon—If configured, this event provides an override behavior to the default account-logon processing. The default behavior only triggers authentication with provided credentials. However, if you override the default account-logon event, then you must explicitly configure the authentication action, and any additional action you require.
- Account-Logoff—If configured, this event provides an override behavior for the default account-logoff processing. The default behavior of the account-logoff processing is to disconnect the subscriber. Being able to override the default behavior is useful. Instead of disconnecting the subscriber, the service provider can perform a re-authentication. The re-authentication is done through a new account-logon by enabling HTTP Redirect feature on the subscriber.
- Idle-Timeout—If configured, this event terminates the IPoE and PPPoE subscriber sessions when the timeout period expires. The default behavior of the Idle-Timeout event is to disconnect the session. You can configure a **monitor** action under the idle timeout event for a subscriber policy, to prevent the termination of the subscriber session when the idle timeout period expires.

Configuring a Policy-Map

Perform this task to configure policy map for control policies. As an example, this policy-map is created for the Session-Start and Session-Activate events. For the Session-Start event, a dynamic template is activated. For the Session-Activate event, an authentication process is invoked. For more information about the supported events, see Control Policy Events, on page 64.

SUMMARY STEPS

1. **configure**
2. **policy-map type control subscriber** *policy-map-name*
3. **event session-start match-all**
4. **class type control subscriber** *class_name* **do-until-failure**
5. *sequence_number* **activate dynamic-template** *dynamic-template_name*

6. **event session-activate match-all**
7. **class type control subscriber** *class_name* **do-until-failure**
8. *sequence_number* **authenticate aaa list default**
9. **end-policy-map**
10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type control subscriber** *policy-map-name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type control subscriber plmap1`	Creates a new policy-map with user-defined name. Enters the policy-map mode.
Step 3	**event session-start match-all** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-all`	Defines an event (session start) for which actions will be performed. Defines the match instruction to be "match-all", which executes actions for all matched classes.
Step 4	**class type control subscriber** *class_name* **do-until-failure** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber CL1 do-until-failure`	Associates a class-map with the event. The class-map name has to be specified. Instructs that the actions will be performed until a failure occurs.
Step 5	*sequence_number* **activate dynamic-template** *dynamic-template_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 1 activate dynamic-template template1`	Defines the action to be performed. In this case, it activates a dynamic-template. **Note** This command can be repeated to define multiple actions.
Step 6	**event session-activate match-all** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# event session-activate match-all`	Defines an event (activate session) for which actions will be performed. Defines the match instruction to be "match-all", which executes actions for all matched classes.
Step 7	**class type control subscriber** *class_name* **do-until-failure** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber CL1 do-until-failure`	Associates a class-map with the event. The class-map name needs to be specified. Instructs that the actions will be performed until a failure occurs.

	Command or Action	Purpose
Step 8	*sequence_number* **authenticate aaa list default** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 2 authenticate aaa list default`	Defines the action to be performed. In this case, it initiates the authentication of AAA list.
Step 9	**end-policy-map** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# end-policy-map`	Ends the policy map configuration.
Step 10	**commit**	

Configuring a Policy-Map: An example

```
policy-map type control subscriber PL1
 event session-start match-first
  class type control subscriber DHCP_class do-until-failure
   1 activate dynamic-template dhcp
  class type control subscriber class-default do-until-failure
! Packet trigger is default
   1 activate dynamic-template packet-trigger
 end-policy-map
 !
!
end

\\Configuring a Policy-Map with idle-timeout event and monitor action

policy-map type control subscriber PL2
  event idle-timeout
   class type control subscriber DHCP_class
    1 monitor
```

Activating Policy-Map

After a policy-map is created, it needs to be activated on a router interface. The policies are implemented only after the policy-map activation is completed. One or more policy-maps will constitute the service-policy. To enable the service-policy, see Enabling a Service-Policy on a Subscriber Interface, on page 67.

Enabling a Service-Policy on a Subscriber Interface

Perform this task to enable a service-policy on a subscriber interface. The process involves attaching a previously created policy-map with an interface. Once this process is complete, the actions defined in the class-map will take effect for the traffic coming on the interface on which service-policy is enabled.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **service-policy type control subscriber** *policy_name*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether100.10`	Enters the interface configuration mode for the bundle-ether access interface. **Note** For IPoE sessions, it is recommended that Dynamic ARP learning be disabled in the access-interface, using the **arp learning disable** command.
Step 3	**service-policy type control subscriber** *policy_name* **Example:** `RP/0/RSP0/CPU0:router(config-if)# service-policy type control subscriber plmap1`	Applies a pre-defined policy-map named 'plmap1' to an access interface.
Step 4	**commit**	

Defining Dynamic Templates

A dynamic template is a container used to group a set of configuration settings, and apply them to the subscriber sessions. A dynamic template is globally configured through CLI. However, defining the dynamic template does not immediately cause the configuration to be applied to a subscriber interface. The configuration within a dynamic template is applied to a subscriber interface, only when the dynamic template is activated using a control policy. Similarly, the applied configurations are stopped, only when the dynamic template is deactivated using a control policy.

There are three basic types of dynamic-templates:

- PPP templates—It contains specific configurations related to the PPPoE protocol.
- IP Subscriber templates—It contains specific configurations that are activated on IP subscriber sessions.
- Service templates—It contains service-related configuration that are activated in response to session life-cycle events. Service templates are precluded from containing interface or media-specific commands.

A dynamic template can either be configured on the CLI, or downloaded from the AAA server. In the following sample configuration, the policy map activates an IP Subscriber dynamic template that is defined on the CLI.

```
dynamic-template
type ipsubscriber ipsub
ipv4 unnumbered Loopback400
```

```
policy-map type control subscriber PL2
event session-start match-first
class type control subscriber class-default do-all
1 activate dynamic-template ipsub
```

There are two types of dynamic templates that are downloaded from the AAA server—user profiles and service profiles. User profiles are applied to a single subscriber, whereas, service profiles can be applied to multiple subscribers. In the following sample configuration, the policy map downloads a service template from the AAA server.

```
Radius Config:
service1 Password="xxxxxx"
Cisco-avpair = "ipv4:ipv4-unnumbered=Loopback400"

Router Config:
policy-map type control subscriber PL2
event session-start match-first
class type control subscriber class-default do-all
1 activate dynamic-template service1 aaa list default
```

In the above example, the "aaa list default" keyword specifies that the template "service1" be downloaded from the AAA server. A template is downloaded only once. If there are multiple control policies referring to service1, then those will get the previously downloaded version.

It is possible to activate more than one dynamic template on the same subscriber interface, for the same event or different events. If the configurations for a particular functionality is defined in multiple dynamic templates, the configurations are derived from all the templates on a certain order of precedence. This order is based on the type of dynamic template, and whether it is being applied from CLI or AAA. The order is:

- Template applied by the user profile from AAA
- Template applied by the service profile from AAA
- IP Subscriber template applied from CLI
- PPP template applied from CLI
- Service template applied from CLI

The tasks involving the use of dynamic templates to define specific feature configurations are included in their corresponding feature topics.

Additional References

These sections provide references related to implementing control policy.

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 5

Establishing Subscriber Sessions

A subscriber accesses network resources through a logical connection known as subscriber session. This chapter provides information about various types of subscriber sessions, namely IPoE and PPPoE, and IP addressing by DHCP.

Table 6: Feature History for Establishing Subscriber Sessions

Release	Modification
Release 4.2.0	Initial release
Release 5.3.0	BNG Subscriber Templates feature was introduced.
Release 5.3.2	Support of Parameterized QoS (PQoS) feature for line card subscribers was added.
Release 5.3.1	Support of Geo Redundancy for PPPoE sessions was added.
Release 5.3.3	Option to prevent default ARP entry creation for a subscriber interface was introduced.
Release 6.0.1	IPv6 router advertisements on IPv4 subscriber interface is introduced.

This chapter covers these topics:

• Additional References, on page 173

Subscriber Session Overview

A session represents the logical connection between the customer premise equipment (CPE) and the network resource. To enable a subscriber access the network resources, the network has to establish a session with the subscriber. Each session establishment comprises of these phases:

Note When packets arrive on an access interface, an attempt is made to link that packet to a subscriber context.

- For PPPoE sessions the Source MAC of the CPE, Access interface and PPPoE Session ID are used to match the remote peer to a subscriber interface.
- For IPoE sessions the Source MAC, Access interface and IP address are verified against the DHCP binding to find a matching subscriber interface.

If there is no match, the packet is mapped against the access (sub-)interface. Considering that the access interface in IPoE designs is IP enabled (eg via an IP-Unnumbered configuration) that packets are processed like regular IP. In order to secure your BNG access interface, you will want to apply either uRPF or an Access-List blocking everything but DHCP incoming on the access interface to limit remote subscribers for which we don't have an interface created from accessing network resources.

- Establishing a connection—in this phase CPE finds the BNG with which to communicate.
- Authenticating and authorizing the subscriber—in this phase, BNG authenticates the subscribers and authorizes them to use the network. This phase is performed with the help of the RADIUS server.
- Giving subscriber an identity—in this phase, the subscriber is assigned an identity, the IP address.
- Monitoring the session—in this phase, BNG ascertains that the session is up and running.

The subscribers are not configured directly on BNG. Instead, a framework is created on which subscriber features and subscriber sessions are started and stopped dynamically. The framework consists of control policies and dynamic templates, which perform these functions:

- Control policy determines the action BNG takes when specific events, such as receipt of a session start request, or failure of authentication, occurs. The action is determined by the class-map defined in the control policy. The action involves activating dynamic templates.
- Dynamic template contains a set of CLI commands that are applied to a subscriber session. Multiple dynamic templates can be activated, one at a time, on the same subscriber interface. Also, the same dynamic template can be activated on multiple subscriber interfaces through different control policies.

Service providers can deploy subscribers over VLAN in these ways:

- 1:1 VLAN model—This model depicts a scenario where one dedicated VLAN is available for each customer. Each VLAN is an q-in-q VLAN where the inner VLAN tag represents the subscriber and the outer VLAN tag represents the DSLAM.
- N:1 VLAN model—This model depicts a scenario where multiple subscribers are available on a shared VLAN. The VLAN tags represent the DSLAM or the aggregation device.

- Ambiguous VLANs —This model allows the operator to specify a large number of VLANs in a single CLI line. Using ambiguous VLAN, a range of inner or outer tags (or both) can be configured on a VLAN sub-interface. This is particularly useful for the 1:1 model, where every subscriber has a unique value for the set of VLAN tags. For more information about ambiguous VLANs, see Subscriber Session on Ambiguous VLANs , on page 239.

The subscriber sessions are established over the subscriber interfaces, which are virtual interfaces. It is possible to create only one interface for each subscriber session. A port can contain multiple VLANs, each of which can support multiple subscribers. BNG creates subscriber interfaces for each kind of session. These interfaces are named based on the parent interface, such as bundle-ether 2.100.pppoe312. The subscribers on bundles (or bundle-VLANs) interfaces allow redundancy, and are managed on the BNG route processor (RP).

For details on subscriber session limit, see Subscriber Session Limit , on page 167.

To provide network redundancy and load balancing, the service provider can deploy multiple links between the DSLAM and the BNG. The individual links can be grouped into ether-bundles, including VLANs over ether-bundles, or link aggregation groups (LAGs). The subscriber sessions can be active on any link within the bundle or group. If a BNG is deployed in a LAG configuration, all traffic for one subscriber should be configured to traverse one link of the ether-bundle. Load-balancing is achieved by putting different subscribers on different links.

There are two mechanisms to establish a subscriber session, namely, IPoE and PPPoE. These are discussed next in the next topics.

Line card (LC) subscribers are supported in BNG. For details, see Line Card Subscribers, on page 163.

BNG supports interface based static sessions, where all traffic belonging to a particular VLAN sub-interface is treated as a single session. For details, see Static Sessions, on page 165.

Note

- If a **clear subscriber session all** command is issued with the intend to clear all the subscriber sessions and if a route processor fail over (RPFO) occurs while the session bring down is in progress, then it is recommended to re-run the same command post RPFO, to ensure all the remaining sessions, if any, are brought down.
- Do not add or delete any Virtual Routing and Forwarding (VRF) configuration when the subscriber sessions are being brought up or brought down. Otherwise, there can be issues while creating new subscriber sessions that can lead to system instability.
- With packet-triggered session initiator configured, new sessions (for subscriber session with already activated state or subscriber sessions which are duplicating the credentials of already activated subscribers) are attempted even before clearing the previous session. This happens while clearing a subscriber session (either using CoA or using **clear subscriber session** command) when the user is sending traffic. From Cisco IOS XR Software Release 5.2.2 and later, if a packet-triggered session gets to an error state (Access-Reject or feature programming error) during session establishment procedure, then a penalty of two minutes is applied to that subscriber. That is, BNG does not accept a new session from the same subscriber for a time period of two minutes. This avoids hogging of system resources by a DoS attack. The penalty remains the same if the session was cleared either using CoA or using clear subscriber session command. For IP-initiated sessions, the subscribers can disconnect either based on the idle timeout or based on the portal logout. For idle timeout scenario, the penalty does not have any impact, because the penalty is applicable only if the subscriber sends traffic while the session is being cleared. In a portal logout scenario, a CoA is triggered by the portal. If subscriber sends traffic when the CoA is received, then the two-minute penalty is applied to that subscriber; else there is no penalty.

  From Cisco IOS XR Software Release 5.3.0 and later, the penalty is reduced to 10 seconds only for scenarios where the previous session of the same subscriber is in **disconnecting** state. For other scenarios, the penalty remains as two minutes.

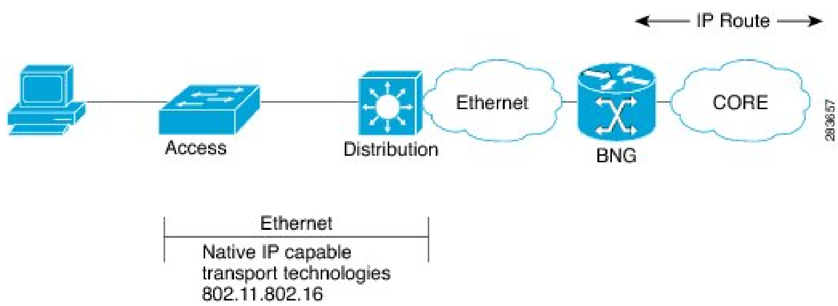
Restrictions

- If the subscriber's VRF is taken from the access interface's VRF value, then the VRF, configured in the dynamic template used by the subscriber, must match. If the two VRFs do not match, then the session would not work properly.
- ACL logging on BNG dynamic template is not supported.

Establishing IPoE Session

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. 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 IP address. A typical IPoE session is depicted in the following figure.

Figure 6: IPoE Session



The process of provisioning an IPoE session involves:

- Enabling the processing of IPv4 or IPv6 protocol on an access interface. See Enabling IPv4 or IPv6 on an Access Interface, on page 76.

  Note For subscriber deployments, it is recommended that Dynamic ARP learning be disabled in the access-interface, using the **arp learning disable** command in the access-interface configuration mode.

- Creating dynamic template that contains the settings for the IPoE sessions. See Creating Dynamic Template for IPv4 or IPv6 Subscriber Session, on page 77.
- Creating policy-map to activate dynamic template. See Creating a Policy-Map to Run During IPoE Session, on page 79.
- Enabling IPoE subscriber creation on access interface by activating service-policy. The service-policy will apply the policy-map on the access interface. See Enabling IPoE Subscribers on an Access Interface, on page 80.

For details on routed subscriber sessions, see Routed Subscriber Sessions, on page 83.

BNG supports IPoE subscriber session-restart. For details, see Subscriber Session-Restart, on page 134.

To limit the default ARP entry creations, see Prevent Default ARP Entry Creation for a Subscriber Interface, on page 92.

Note If an access interface in BNG is configured to support only packet (PKT) triggered sessions, or both DHCP and PKT triggered sessions, then a burst of traffic with unique flows can affect the BNG router in terms of processing each packet to determine if it is an IPoE (PKT triggered) packet. New subscriber sessions cannot be established in these scenarios and this can in turn lead to system instability. Therefore, it is mandatory to configure static lpts policer for **unclassified rsp** protocol, on each of the line cards (LCs), such that the traffic rate does not exceed 150 pps per LC. The rate configured is applied at network processor (NP). Therefore, for an LC with 4 NPs, the rate should be configured as 38 (150/4), to achieve a traffic rate of 150 pps. For example, `lpts punt police location 0/RSP0/CPU0 protocol unclassified rsp rate 38`.

Restrictions

Enabling IPoE subscribers on an access-interface is subjected to a restriction that packet-triggered L2 sessions (**initiator unclassified-source**) are not supported for IPv6.

Enabling IPv4 or IPv6 on an Access Interface

Perform these tasks to enable IPv4 and IPv6 processing on an access interface. In this example, the IPv4 is being provisioned on an unnumbered bundle-interface.

SUMMARY STEPS

1. **configure**
2. **interface** *type* *interface-path-id*
3. **arp learning disable**
4. **ipv4 unnumbered** *interface-type* *interface-instance*
5. **ipv6 enable**
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type* *interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether100.10`	Enters interface configuration mode for the bundle-interface.
Step 3	**arp learning disable** **Example:** `RP/0/RSP0/CPU0:router(config-if)# arp learning disable`	Disables arp learning for the access-interface.
Step 4	**ipv4 unnumbered** *interface-type* *interface-instance* **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered loopback 5`	Enables IPv4 processing on a unnumbered interface without assigning an explicit IPv4 address to that interface. Instead, the IP address is borrowed from the loopback interface. For the "ipv4 unnumbered" command, you must specify another interface in the same router that has been assigned an IP address and whose status is displayed as “up” for the **show interfaces** command.
Step 5	**ipv6 enable** **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv6 enable`	Enables IPv6 processing on an unnumbered interface that has not been assigned an explicit IPv6 address. **Note** This step not only enables IPv6 processing on the interface, but also assigns an IPv6 link-local unicast address to it.

	Command or Action	Purpose
Step 6	**commit**	

Enabling IPv4 or IPv6on an Access Interface: Examples

```
//Enabling IPv4 on an Access Interface

configure
interface Bundle-Ether100.10
arp learning disable
ipv4 unnumbered loopback 5
!
!
end


//Enabling IPv6 on an Access Interface

configure
interface Bundle-Ether100.10
arp learning disable
ipv6 enable
!
!
end
```

Creating Dynamic Template for IPv4 or IPv6 Subscriber Session

Perform this task to create a dynamic template for IPv4 or IPv6 subscriber session. As an example, in this dynamic template you will specify the MTU value for the IPv4 or IPv6 session and enable uRPF. The uRPF ensures that the traffic from malformed or forged IPv4 source addresses are not accepted on the subscriber interface. For more information about uRPF feature, see uRPF, on page 244.

SUMMARY STEPS

1. **configure**
2. **dynamic-template**
3. **type** { **ipsubscriber** | **ppp** | **service** } *dynamic-template-name*
4. **timeout idle** *value* [**threshold** *duration*] [**traffic** {**both** | **inbound** | **outbound**}]
5. **accounting aaa list default type session periodic-interval** *value* **dual-stack-delay** *value*
6. {**ipv4** | **ipv6**} **mtu** *mtu-bytes*
7. {**ipv4** | **ipv6**}**verify unicast source reachable-via rx**
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**dynamic-template** **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template`	Enters the dynamic-template configuration mode.
Step 3	**type** { **ipsubscriber** | **ppp** | **service** } *dynamic-template-name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ipsubscriber ipsub1`	Creates a dynamic-template with an user-defined name for IP subscriber.
Step 4	**timeout idle** *value* [**threshold** *duration*] [**traffic** {**both** | **inbound** | **outbound**}] **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# timeout idle 600 threshold 2 traffic both`	IPv4 or IPv6 or Dual-stack Subscribers support idle timeout feature. **Note** You can configure a **monitor** action under the idle timeout event for a subscriber policy, to prevent the termination of subscriber sessions when the idle timeout period expires.
Step 5	**accounting aaa list default type session periodic-interval** *value* **dual-stack-delay** *value* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# accounting aaa list default type session periodic-interval 60 dual-stack-delay 1`	Configures the subscriber accounting feature.

Step 6	{**ipv4** | **ipv6**} **mtu** *mtu-bytes*
	Example:
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv4 mtu 678`
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 mtu 548`
		Sets IPv4 or IPv6 maximum transmission unit (MTU). The range is from 68 to 65535 bytes. The MTU value defines the largest packet size that can be transmitted during the subscriber session.

Step 7	{**ipv4** | **ipv6**}**verify unicast source reachable-via rx**
	Example:
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv4 verify unicast source reachable-via rx`
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 verify unicast source reachable-via rx`
		Enables uRPF for packet validation that performs source address reachability check.

Step 8	**commit**	

Creating Dynamic Template for IPv4 or IPv6 Subscriber Session: Examples

```
//Creating Dynamic Template for IPv4 Subscriber Session

configure
dynamic-template
type ipsubscriber ipsub1
timeout idle 600
accounting aaa list default type session periodic-interval 60 dual-stack-delay 1
ipv4 mtu 678
ipv4 verify unicast source reachable-via rx
!
!
end
```

```
//Creating Dynamic Template for IPv6 Subscriber Session

configure
dynamic-template
type ipsubscriber ipsub1
timeout idle 600 threshold 2 traffic both
accounting aaa list default type session periodic-interval 60 dual-stack-delay 1
ipv6 mtu 678
ipv6 verify unicast source reachable-via rx
!
!
end
```

Creating a Policy-Map to Run During IPoE Session

Perform this task to create a policy-map that will activate a predefined dynamic-template during an IPoE subscriber session. As an example, this policy-map activates a dynamic template, and applies a locally defined authorization setting, during a session-start event.

SUMMARY STEPS

1. **configure**
2. **policy-map type control subscriber** *policy_name*
3. **event session-start match-first**
4. **class type control subscriber** *class_name* **do-until-failure**
5. *sequence_number* **activate dynamic-template** *dynamic-template_name*
6. *sequence_number* **authorize aaa list default format** *format_name* **password** *password*
7. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type control subscriber** *policy_name* **Example:**	Creates a new policy map of the type "control subscriber" with the name "IPoE_policy".

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# policy-map type control subscriber IPoE_policy`	
Step 3	**event session-start match-first** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-first`	Defines an event (session start) for which actions will be performed.
Step 4	**class type control subscriber** *class_name* **do-until-failure** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber class-default do-until-failure`	Configures the class to which the subscriber has to be matched. When there is a match, executes all actions until a failure is encountered.
Step 5	*sequence_number* **activate dynamic-template** *dynamic-template_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 1 activate dynamic-template ipsub1`	Allows authentication of the subscriber to be triggered using the complete structure username.
Step 6	*sequence_number* **authorize aaa list default format** *format_name* **password** *password* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 1 authorize aaa list default format RM_User password Cisco`	Allows authorization of the subscriber to be triggered using the domain name of the subscriber. Also, provides domain format-rule, which helps to parse the domain from a complete structured username.
Step 7	**commit**	

Creating a Policy-Map to Run During IPoE Session: An example

```
configure
policy-map type control subscriber IPoE_policy
event session-start match-first
class type control subscriber class-default do-until-failure
1 activate dynamic-template ipsub1
1 authorize aaa list default format RM_User password Cisco
!
!
end
```

Enabling IPoE Subscribers on an Access Interface

Perform this task to enable IPoE subscriber creation on an access interface.

SUMMARY STEPS

1. **configure**
2. **interface** *interface-type* *interface-path-id*
3. **arp learning disable**
4. {**ipv4** | **ipv6**} **address** {*ipv4_address* | *ipv6_address*} *ipsubnet_mask*
5. **service-policy** **type control subscriber** *policy-name*
6. **encapsulation** **dot1q** *value*
7. **ipsubscriber** {**ipv4** | **ipv6**}**l2-connected**
8. **initiator** **dhcp**
9. **initiator** **unclassified-source [address-unique]**
10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**interface** *interface-type* *interface-path-id*
	Example:
	`RP/0/RSP0/CPU0:router(config)# interface Bundler-Ether400.12`
		Configures an interface and enters interface configuration mode.
		• The type argument specifies an interface type. For more information on interface types, use the question mark (?) online help function.
		• The instance argument specifies either a physical interface instance or a virtual instance.
		  • The naming notation for a physical interface instance is rack/slot/module/port. The slash (/) between values is required as part of the notation.
		  • The number range for a virtual interface instance varies depending on the interface type.

Step 3	**arp learning disable** **Example:** `RP/0/RSP0/CPU0:router(config-if)# arp learning disable`	Disables arp learning for the access-interface.

Step 4	{**ipv4** | **ipv6**} **address** {*ipv4_address* | *ipv6_address*} *ipsubnet_mask*
	Example:
	`RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 3.5.1.1 255.255.0.0`
	or
	Example:
	`RP/0/RSP0/CPU0:router(config-subif)# ipv6 address 1144:11`
		Sets the IPv4 address or an IPv6 address for an interface.

	Command or Action	Purpose
Step 5	**service-policy type control subscriber** *policy-name* **Example:** `RP/0/RSP0/CPU0:router(config-subif)# service-policy type control subscriber PL4`	Associates a subscriber control service policy to the interface. **Note** Refer to the "Configuring a Subscriber Policy Map" procedure to create a PL4 policy-map.
Step 6	**encapsulation dot1q** *value* **Example:** `RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 40`	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance. The value ranges from 1 to 4094.

Step 7	**ipsubscriber** {**ipv4** | **ipv6**}**l2-connected**
	Example:
	`RP/0/RSP0/CPU0:router(config-subif)# ipsubscriber ipv4 l2-connected`
	or
	Example:
	`RP/0/RSP0/CPU0:router(config-subif)# ipsubscriber ipv6 l2-connected`
		Enables creations of L2-connected IPv4 or IPv6 subscribers on the sub-interface.
		Note It is not recommended to remove these call flow-initiated configurations, after subscriber sessions are active:
		• For an IPoE subscriber session, you must not delete the **ipsubscriber ipv4 l2-connected initiator dhcp** command from the sub-interface
		• For a packet-triggered subscriber session, you must not delete the **ipsubscriber ipv4 l2-connected initiator unclassified-source** command from the sub-interface.

Step 8	**initiator dhcp**
	Example:
	`RP/0/RSP0/CPU0:router(config-subif-ipsub-ipv4-l2conn)# initiator dhcp`
	or
	Example:
	`RP/0/RSP0/CPU0:router(config-subif-ipsub-ipv6-l2conn)# initiator dhcp`
		Configures DHCP as the first-sign-of-life (FSOL) protocol for IP subscriber.

Step 9	**initiator unclassified-source [address-unique]** **Example:** `RP/0/RSP0/CPU0:router(config-subif-ipsub-ipv4-l2conn)# initiator unclassified-source`	Configures unclassified packets as the first-sign-of-life (FSOL) for IPv4 subscriber. The **address-unique** option enables subscriber IP uniqueness check during FSOL processing, thereby preventing invalid sessions from creating interfaces. This option is available from Cisco IOS XR Software Release 5.2.2 and later.

	Command or Action	Purpose
		Note • The **initiator unclassified-source** option is not supported for IPv6. • If multiple initiators are used, use a policy or class map to prevent overlap of the IP addresses for the different sources.
Step 10	**commit**	

Enabling IPoE Subscribers on an Access Interface: Examples

```
configure
interface Bundler-Ether400.12
arp learning disable
ipv4 address 3.5.1.1 255.255.0.0
service-policy type control subscriber PL4
encapsulation dot1q 40
ipsubscriber ipv4 l2-connected
initiator dhcp
initiator unclassified-source
!
!
end
```

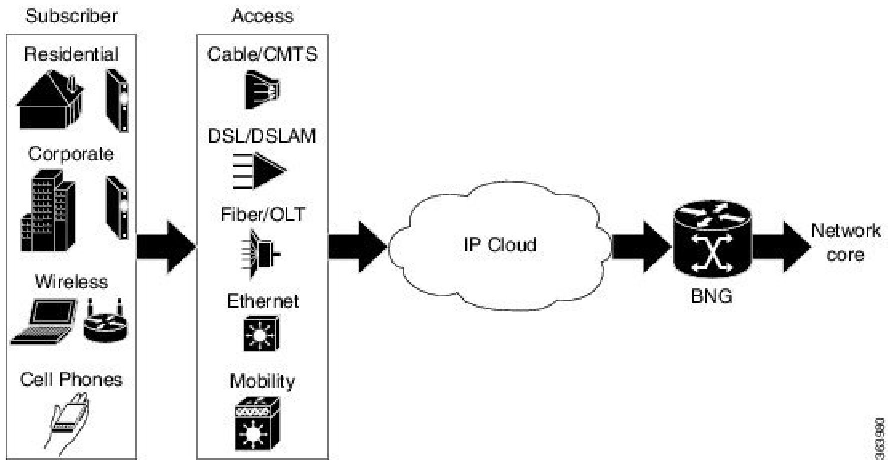
```
configure
interface Bundler-Ether400.12
arp learning disable
ipv6 address 4444:34
service-policy type control subscriber PL4
encapsulation dot1q 40
ipsubscriber ipv6 l2-connected
initiator dhcp
!
!
end
```

Routed Subscriber Sessions

BNG supports L3 or routed subscriber sessions (DHCP-initiated and Packet-triggered), where IP subscribers are connected through a routed access network. The policies and services on the routed subscriber sessions are applied in a similar manner as with L2 subscriber sessions.

This figure shows a typical routed subscriber session network model:

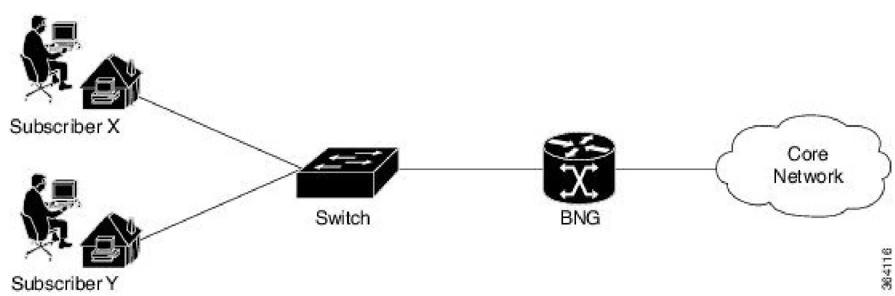
Figure 7: Routed Subscriber Session Network Model



L2-connected subscribers are either directly attached to the physical interfaces of BNG or connected to BNG through a Layer 2 access network, such as a bridged or a switched network. Each user device here is a unique subscriber session. In case there is a routed CPE, the CPE owns the subscriber session on the BNG, and all devices behind the CPE performs NAT. The CPE holds the start of the session to BNG. The subscriber is keyed on the MAC address. Because there is a switched network, the BNG directly sees the MAC address of the device.

This figure shows a typical L2-connected subscriber session:

Figure 8: L2-connected Subscriber Session



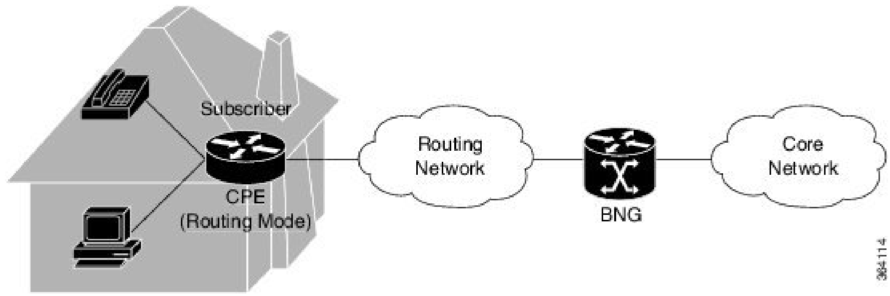
Whereas, routed subscribers are connected to BNG through routed device(s). The devices behind the CPE's MAC address are not visible to BNG. The subscriber is no longer keyed on MAC address. Instead, ip-address is used to key the subscriber session of the device.

In a typical L3 routed aggregation model, the CPE uses NAT to cover up the devices behind the routed CPE. The BNG sees a subscriber session that is initiated and linked to the WAN interface of the routed CPE.

With this routed subscriber session functionality, you can connect devices and create subscriber sessions that are behind a routed CPE.

This figure shows a typical routed subscriber session:

Figure 9: Routed Subscriber Session



To configure an access-interface to host routed subscriber sessions, see Configuring Routed Subscriber Sessions, on page 90.

Routed subscriber sessions come up only if a summary route is added on BNG. The summary route can be either statically configured, or created through some of the routing protocols like OSPF or EIGRP. The summary route VRF must be same as the access-interface VRF in BNG. Modifying or deleting a summary route that is pointing to the subscriber access-interface, while the subscriber sessions are active, may cause a minimal traffic disruption due to route re-convergence. Therefore, it is recommended that the summary route pointing to the subscriber access-interface be modified or deleted only after deleting the sessions that are using that static summary route.

DHCP-initiated Routed Subscriber Sessions

BNG supports DHCPv4-initiated routed subscriber sessions.

DHCP Interaction

The DHCP pool IP address range in BNG must be in compliance with the summary route address range. This DHCP pool IP address range must also match the IP address subnet of the first hop router, which acts as the DHCP relay or proxy. The route for this particular address range must be configured in BNG, so that BNG can reach the subnet of the first hop router, and eventually reach the subscriber.

The subscriber route need not be explicitly added. It is added internally by the BNG process, when the subscriber session is up.

For routed subscriber sessions, the DHCP server should be configured locally on ASR9K router itself, or a DHCP radius proxy should be used. Proxy mode to an external DHCP server is not supported. For details on the call flow of a DHCPv4-initiated session, see Call Flow of DHCPv4-initiated Routed Subscriber Sessions, on page 86.

Session Initiator and Session Identifier

Routed sessions should use IP-based session in-band initiator; whereas L2 connected sessions can have **unclassified-mac** as session in-band initiator. Only DHCPv4 initiated sessions are supported.

Access Interface Features

Although features like ACL and Netflow may be configured on the access-interface, they do not get applied on the subscriber traffic under the respective access-interface. Which features get applied on the subscriber interface is decided based on the dynamic-template configurations under the interface or through RADIUS profile.

VRF Mapping

Routed subscriber sessions support VRF mapping, which allows subscriber to be in a different VRF other than the access-interface VRF. The DHCP pool VRF in BNG must be same as the subscriber VRF, whereas the summary route VRF must be same as the access-interface VRF in BNG. During subscriber creation, information from the dynamic-template or RADIUS is used to set the subscriber VRF. Because access-interface is not used to classify subscriber traffic, the IP address given to subscriber in a given access-interface must be a non-overlapping address.

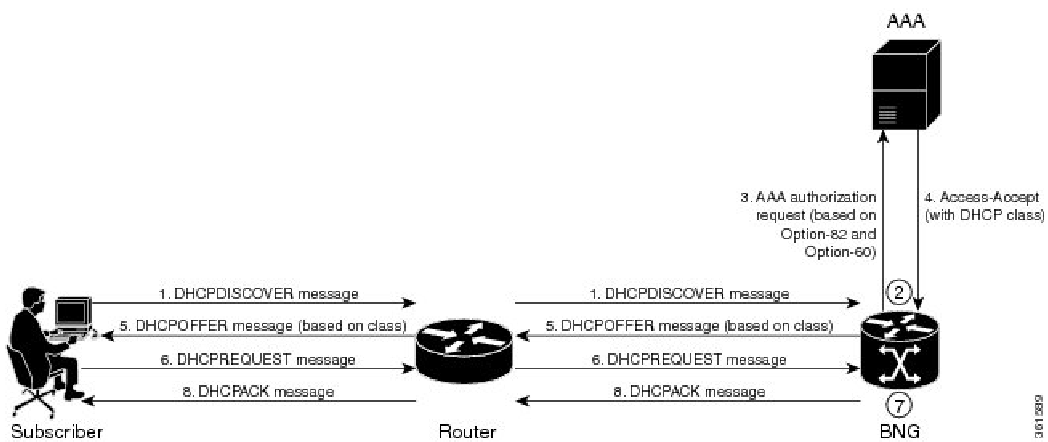
Non-Subscriber Traffic

Because DHCP is the only session initiator for a routed subscriber, a non-subscriber packet is routed as a normal packet on an access-interface. For such packets, the features on access interface are applicable as normal. To prevent such traffic, you should deploy ACL on the access interface.

Call Flow of DHCPv4-initiated Routed Subscriber Sessions

This figure shows a call flow of DHCPv4-initiated routed subscriber session:

Figure 10: Call Flow of DHCPv4-initiated Routed Subscriber Session



These are the detailed steps involved in the DHCPv4 call flow :

1. The subscriber connects to the network and sends a DHCPDISCOVER broadcast packet on the network. The first hop router, configured as a DHCP relay or a DHCP proxy, processes the DHCPDISCOVER message and unicasts it to the BNG that acts as a DHCP server.
2. The BNG creates the subscriber session in its policy plane, and executes the policy rules on the session.
3. As per the policy rule, the BNG sends an AAA authorization request based on Option-82 and Option-60 to the RADIUS server.
4. The RADIUS server replies to the BNG with an Access-Accept message containing DHCP class information that is used for the subscriber IP address assignment.
5. The DHCP server on the BNG uses the DHCP class information in the Access-Accept message to allocate an IP address from an appropriate address pool, and sends a DHCPOFFER message to the subscriber.
6. The subscriber accepts the IP address and sends a DHCPREQUEST message back to the BNG.
7. The BNG assigns IPv4 address to the subscriber; from this point onwards, the session on the BNG starts accepting traffic from the subscriber.

8. The BNG sends a DHCPACK message to the subscriber.

The first hop router can act as either a DHCP relay or a DHCP proxy. In the case of a DHCP proxy, the first hop router maintains the DHCP binding, and it also acts as a DHCP server to the subscriber.

When a DHCP binding is deleted, the BNG session associated with it is also deleted. Because DHCPv4 is the only session initiator, IP address changes cannot happen without having the DHCP server run on BNG. Therefore, in the case of an IP address change, the DHCP deletes the previous session and creates a new session.

Packet-triggered Routed Subscriber Sessions

BNG supports packet-triggered IPv4 and IPv6 routed subscriber sessions. Also, packet-triggered FSOL IPv4 and IPv6 on the same access interface are supported.

Note This feature is available from Cisco IOS XR Software Release 5.2.2 and later.

Session Initiator and Session Identifier

The **unclassified-ip** is used as the initiator for packet-triggered IPv4 and IPv6 routed subscriber sessions. The routed session is identified by subscriber-prefix and prefix-length.

In the case of dual-stack (that is, if both address-families are enabled on a CPE), two separate sessions are created on BNG - one for IPv4 and another for IPv6. Also, if RADIUS profile has dual-stack configuration, the entire configuration does not take effect; only the profile for the address-family takes effect.

The access interface features and VRF mapping for packet-triggered routed subscriber sessions remain the same as that for DHCP-initiated sessions.

For IPv6 packet-triggered routed subscribers, you can perform CoA using session identifier as standard **Framed-IPv6-Prefix** or AVpair **addrv6** RADIUS attribute.

Configuring Packet-triggered Routed Subscriber Sessions

This command is configured on the access interface, to make all the subscribers coming on that interface to routed subscribers:

```
ipsubscriber ipv4 routed
  initiator unclassififed-ip

ipsubscriber ipv6 routed
  initiator unclassififed-ip [prefix-len]
```

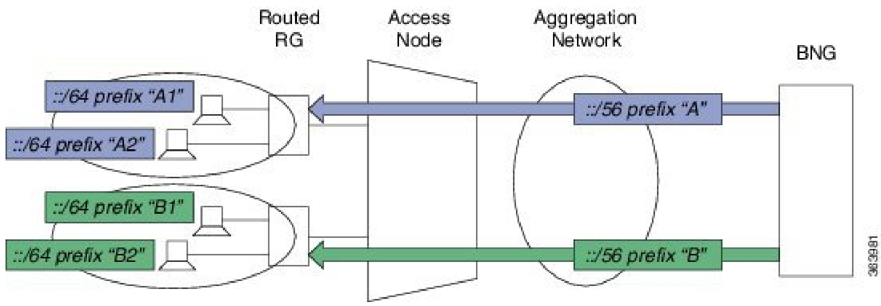
Here, *prefix-len* is the prefix-length of subscriber route. By default, this value is 32 and 128 for IPv4 and IPv6 subscribers respectively.

For a sample deployment topology and use-case scenario of packet-triggered routed subscriber sessions, see Routed Subscriber Deployment Topology and Use Cases, on page 382.

Deployment Model for IPv6 Routed Network

This figure depicts a typical TR-177 routed IPv6 residential gateway deployment:

Figure 11: TR-177 Routed IPv6 Residential Gateway Deployment



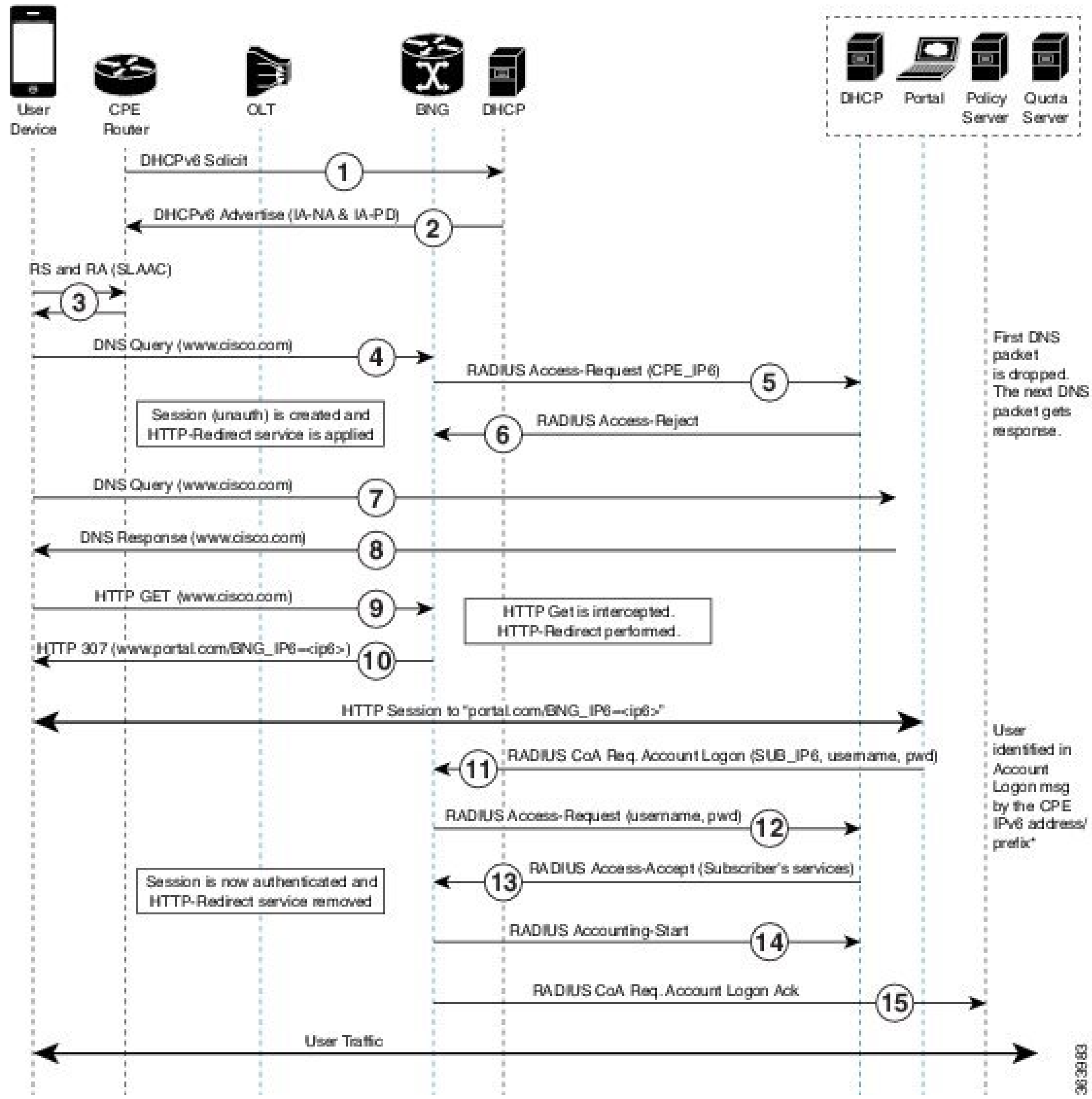
Here, the BNG router acts as a DHCPv6 server, proxy or relay (the DHCP functions can be off-box also) with IA-NA and IA-PD option enabled. BNG allocates both IA-NA and IA-PD non-shared different prefix (/56) for different access networks. Routed Residential Gateway (RG) uses the IA-NA address for itself. Again, Routed RG uses the IA-PD prefix (/56) to distribute different delegated prefixes (/64) to different LAN segments attached to it, using SLAAC or DHCPv6. When the end subscriber starts sending packets, the subscriber session is triggered on BNG.

In another deployment scenario, CPEs with /128 prefix-Len are terminated on BNG. Here, each subscriber is individually authenticated on BNG.

Call Flow of IPv6 Routed Subscriber Session

This figure depicts a typical call flow of web-logon packet-triggered IPv6 routed subscriber session in BNG:

Figure 12: Call Flow of Web-logon IPv6 Routed Subscriber Session



Restrictions for Routed Subscriber Sessions

Support for BNG routed subscriber sessions is subjected to these restrictions:

- Overlapping IP addresses are not supported on the same access-interface.
- Overlapping IP addresses are not supported on the same VRF.
- DHCP-initiated IPv6 sessions are not supported.
- Dual-stack sessions are not supported.
- DHCP lease query is not supported.
- Line card subscribers are not supported.
- For IPv4, BNG cannot be used as DHCP server or proxy to lease IPv4 addresses to IPv4-routed packet-triggered subscribers.

- For IPv6, on-box DHCPv6 server or DHCPv6 proxy can be used to lease IPv6 PD addresses to CPE; but not to end subscribers.
- Because Neighbor Discovery (ND) is point-to-point, ND-triggered sessions (Router Solicitation) are not supported.
- A maximum of only 1 ECMP path is supported for each covering route.

Configuring Routed Subscriber Sessions

Perform this task to configure routed subscriber sessions on an access-interface:

Before you begin

Configuring routed subscriber session in BNG is subjected to these guidelines:

- You must configure dynamic or static routes on the router for subscriber IP addresses. These routes should be configured in such a way that they are synchronized with the way DHCP assigns the IP addresses.

For DHCP-initiated sessions:

- To authorize the subscriber on session-start, you must configure policy-map with a policy having Option-82 (**circuit-id** and **remote-id**) and Option-60 as identifiers.

For Packet-triggered sessions:

- While creating the route (also called as cover route), the route prefix must be smaller than the subscriber prefix. Else, the subscriber route does not install, and the session fails. The cover route must be installed in the access-vrf.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ipsubscriber** {**ipv4** | **ipv6**} **routed**
4. **initiator** {**dhcp** | **unclassified-ip** [**prefix-len** *prefix-len*]}
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface bundle-ether101.201`	Specifies an access-interface and enters the interface configuration mode.

	Command or Action	Purpose
Step 3	**ipsubscriber** {**ipv4** | **ipv6**} **routed** **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipsubscriber ipv4 routed`	Configures the access-interface to accept routed subscriber sessions.

Step 4	**initiator** {**dhcp** | **unclassified-ip** [**prefix-len** *prefix-len*]}
	Example:
	`RP/0/RSP0/CPU0:router(config-if)# initiator dhcp`
	or
	`RP/0/RSP0/CPU0:router(config-if)# initiator unclassified-ip`
		Configures the session initiator as DHCP or unclassified-ip, for routed subscribers.
		Note • DHCP-initiated IPv6 sessions are not supported.
		• **prefix-len** option is applicable only for packet-triggered (initiator unclassified-ip) IPv6 sessions.

Step 5	**commit**	

Configuring Routed Subscriber Sessions: An example

DHCP-initiated routed subscriber sessions:

```
 interface Bundle-Ether101.201
 vrf vpn1
 ipv4 address 10.1.1.1 255.255.255.0
 service-policy type control subscriber ROUTED_POLICY
 encapsulation dot1q 201 second-dot1q 301
 ipsubscriber ipv4 routed
  initiator dhcp
 !
!

//Configuring static summary route
!
router static
 address-family ipv4 unicast
  14.0.0.0/16 12.0.0.2
 !

//Configuring DHCP address pool
!
pool vrf default ipv4 ROUTED_POOL1
 network 14.0.0.0/16
 exclude 14.0.0.1 0.0.0.0
!
```

Packet-triggered routed subscriber sessions:

```
interface Bundle-Ether1.201
 ipv4 address 15.15.15.1 255.255.255.0
 ipv6 address 15:15:15::1/64
 service-policy type control subscriber PL
 encapsulation dot1q 201
```

```
 ipsubscriber ipv4 routed
  initiator unclassified-ip
 !
 ipsubscriber ipv6 routed
  initiator unclassified-ip
 !
!
```

Prevent Default ARP Entry Creation for a Subscriber Interface

In certain deployment scenarios, the subscriber access-interfaces are unnumbered and the associated loopback interface may have multiple secondary IP addresses. These unnumbered interfaces inherit all attributes, including the secondary IP addresses, from the loopback interface. This creates multiple local ARP entries per subscriber interface and the ARP table may extend beyond the supported scale in such scenarios. You can now prevent such default ARP entry creations by using the **subscriber arp scale-mode-enable** command. This functionality does not impact the existing ARP behavior for the subscribers.

Configuration Example

```
Router(config)# subscriber arp scale-mode-enable
```

Establishing PPPoE Session

The PPP protocol is mainly used for communications between two nodes, like a client and a server. The PPP protocol provides a standard method for transporting multi-protocol diagrams 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 can use the LCP to negotiate various link layer properties or characteristics. The NCP is used to establish and configure the associated network protocol before data packets for the protocol can be transmitted.

One of the methods to establish PPP connection is by the use of PPP over Ethernet (PPPoE). In a PPPoE session, the Point-to-Point (PPP) protocol runs between the CPE and BNG. The Home Gateway (which is part of the CPE) adds a PPP header (encapsulation) that is terminated at the BNG.

CPE detects and interacts with BNG using various PPPoE Active Discovery (PAD) messages listed here:

- PPPoE Active Discovery Initiation (PADI)—The CPE broadcasts to initiate the process to discover BNG.
- PPPoE Active Discovery Offer (PADO)—The BNG responds with an offer.
- PPPoE Active Discovery Request (PADR)—The CPE requests to establish a connection.
- PPPoE Active Discovery Session confirmation (PADS)—BNG accepts the request and responds by assigning a session identifier (Session-ID).
- PPPoE Active Discovery Termination (PADT)—Either CPE or BNG terminates the session.

In redundant BNG setups, where the PPPoE client is connected to multiple BNGs, the PADI message sent by the CPE is received on all BNGs. Each BNG, in turn, replies with a PADO message. You must configure Smart Server Selection on BNG to allow subscribers to select one of the BNGs in a multi-BNG setup. Refer PPPoE Smart Server Selection, on page 112

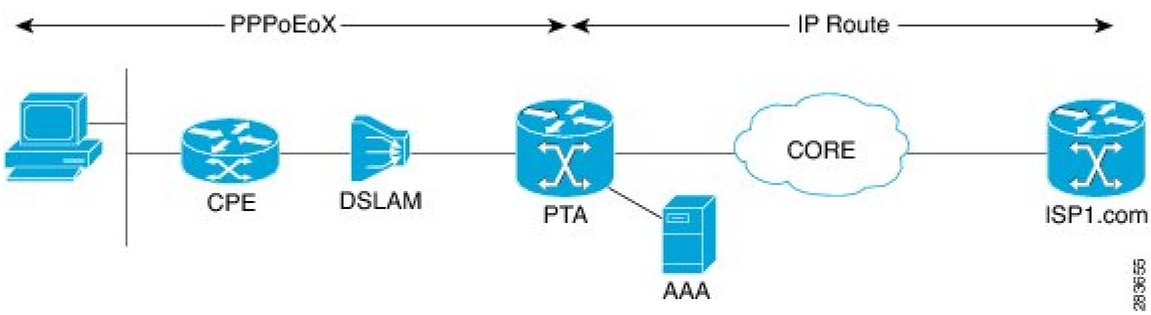
The BNG provides configuration flexibility to limit and throttle the number of PPPoE sessions requests, based on various parameters. For details, see PPPoE Session Limit, on page 114 and PPPoE Session Throttle, on page 116.

The PPPoE session are of two types, PPP PTA and PPP LAC. For the functioning of PPP PTA and PPP LAC session, the RADIUS server must be set up to authenticate and forward sessions as necessary. There is no local authentication available on BNG. The PPP PTA and PPP LAC sessions are explained in the sections, Provisioning PPP PTA Session, on page 93 and Provisioning PPP LAC Session, on page 99.

Provisioning PPP PTA Session

In a PPP Termination and Aggregation (PTA) session, the PPP encapsulation is terminated on BNG. After it is terminated, BNG routes the traffic to the service provider using IP routing. A typical PTA session is depicted in this figure.

Figure 13: PTA Session



PPPoE session configuration information is contained in PPPoE profiles. After a profile has been defined, it can be assigned to an access interface. Multiple PPPoE profiles can be created and assigned to multiple interfaces. A global PPPoE profile can also be created; the global profile serves as the default profile for any interface that has not been assigned a specific PPPoE profile.

The PPP PTA session is typically used in the Network Service Provider (retail) model where the same service operator provides the broadband connection to the subscriber and also manages the network services. The process of provisioning a PPP PTA session involves:

- Creating a PPPoE profile for PPPoE session. See, Creating PPPoE Profiles, on page 93.
- Creating dynamic template that contains the various settings for the PPPoE sessions. See, Creating a PPP Dynamic-Template, on page 94.
- Creating policy-map to activate the dynamic template. See, Creating a Policy-Map to Run During PPPoE Session, on page 95.
- Enabling subscriber creation, and apply the PPPoE profile and service-policy on the access interface. See, Applying the PPPoE Configurations to an Access Interface, on page 98.

  The subscriber creation function must be explicitly enabled on BNG. Unless this function is enabled, the system will not attempt subscriber classification. As a result, the packets get forwarded based on the incoming interface mode.

Creating PPPoE Profiles

Perform this task to create PPPoE profiles. The PPPoE profile will later be applied to an access interface.

SUMMARY STEPS

1. **configure**
2. **pppoe bba-group** *bba-group name*
3. **service name** *service_name*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**pppoe bba-group** *bba-group name* **Example:** `RP/0/RSP0/CPU0:router(config)# pppoe bba-group bba_1`	Creates a PPPoE profile with an user-specified name.
Step 3	**service name** *service_name* **Example:** `RP/0/RSP0/CPU0:router(config-bbagroup)# service name service_1`	Indicates the service that is requested by the subscriber. Repeat this step for each service name that you want to add to the subscriber profile.
Step 4	**commit**	

Creating PPPoE Profiles: An example

```
configure
pppoe bba-group bba_1
service name service_1
!
!
end
```

Creating a PPP Dynamic-Template

Perform this task to create a PPP dynamic-template. As an example, this dynamic-template is created to apply PAP and CHAP authentication methods.

SUMMARY STEPS

1. **configure**
2. **dynamic-template type ppp** *dynamic_template_name*
3. **ppp authentication pap**
4. **ppp authentication chap**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dynamic-template type ppp** *dynamic_template_name* **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template type ppp ppp_pta_template`	Creates a dynamic-template with user-defined name for PPP session.
Step 3	**ppp authentication pap** **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ppp authentication pap`	Enables the use of PAP type authentication during link negotiation by Link Control Protocol (LCP).
Step 4	**ppp authentication chap** **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ppp authentication chap`	Enables the use of CHAP type authentication during link negotiation by Link Control Protocol (LCP).
Step 5	**commit**	

Creating a PPP Dynamic-Template: An example

```
configure
dynamic-template type ppp ppp_pta_template
ppp authentication pap
ppp authentication pap chap
!
!
end
```

Creating a Policy-Map to Run During PPPoE Session

Perform this task to create a policy-map that will activate a PPP dynamic-template during a PPPoE subscribers session. As an example, this policy-map activates a dynamic template during a session-start event. Also, this policy-map applies a locally-defined authorization setting during a session-activate event.

SUMMARY STEPS

1. **configure**
2. **policy-map type control subscriber** *policy_name*
3. **event session-start match-all**
4. **class type control subscriber** *class_name* **do-until-failure**
5. *sequence_number* **activate dynamic-template** *dynamic-template_name*
6. **event session-activate match-all**
7. **class type control subscriber** *class_name* **do-until-failure**

8. *sequence_number* **authenticate aaa list default**
9. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type control subscriber** *policy_name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type control subscriber PPPoE_policy`	Creates a new policy map of the type "control subscriber" with the user-defined name "PPPoE_policy".
Step 3	**event session-start match-all** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-all`	Defines an event (session start) for which actions will be performed.
Step 4	**class type control subscriber** *class_name* **do-until-failure** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber pta_class do-until-failure`	Configures the class to which the subscriber is to be matched. When there is a match, executes all actions until a failure is encountered.
Step 5	*sequence_number* **activate dynamic-template** *dynamic-template_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 1 activate dynamic-template ppp_pta_template`	Activates the dynamic-template with the specified dynamic template name.
Step 6	**event session-activate match-all** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# event session-activate match-all`	Defines an event (session activate) for which actions are performed.
Step 7	**class type control subscriber** *class_name* **do-until-failure** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber PPP_class do-until-failure`	Configures the class to which the subscriber is to be matched. When there is a match, executes all actions until a failure is encountered.
Step 8	*sequence_number* **authenticate aaa list default** **Example:**	Allows authentication of the subscriber to be triggered using the complete structure username.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-pmap-c)# 1 authenticate aaa list default`	
Step 9	**commit**	

Creating a Policy-Map to Run During PPPoE Session: An example

```
configure
policy-map type control subscriber policy1
event session-start match-all
class type control subscriber pta_class do-until-failure
1 activate dynamic-template template1
!
!
event session-activate match-all
class type control subscriber pta_class1 do-until-failure
1 activate dynamic-template ppp_pta_template
end-policy-map
```

Modifying VRF for PPPoE Sessions

BNG does not support modification of VRF using single dynamic template activated on session start. In order to change the VRF for PPPoE sessions from RADIUS, you must split the dynamic template. One dynamic template must be activated in session-start (for PPP parameters). The other dynamic template must contain L3 parameters and it must be enabled on session-activate event after the authenticate step.

This example shows a sample dynamic template configuration and a policy-map configuration for such a VRF transfer scenario, where some PPPoE users must be terminated in a different VRF than the normal user VRF. In order to do so, the user sends two AV-Pairs through RADIUS.

```
dynamic-template
type ppp PPP_TPL                    ===> Layer 3 interface
ppp authentication chap
ppp ipcp peer-address pool IPv4
ipv4 unnumbered Loopback100         ===> Loopback in Global Routing Table
type ppp PPP_TPL_NO_LO              ===> Layer 2 interface
ppp authentication chap

policy-map type control subscriber BNG_PPPOE
event session-activate match-first
class type control subscriber PPP do-until-failure
10 authenticate aaa list default
 20 activate dynamic-template PPP_TPL
event session-start match-first
class type control subscriber PPP do-until-failure
10 activate dynamic-template PPP_TPL_NO_LO
```

Here, the Layer 2 dynamic template is created first, and only PPP authentication is done on it. Therefore, the RADIUS request is sent. The RADIUS returns the attributes and then the BNG proceeds to the next step, that is, session-activate. In session-activate, another dynamic template interface which has layer 3 configuration is used. But, because the BNG has already received the RADIUS attribute for the user, it uses the ipv4 unnumbered contained in the RADIUS profile, rather than the one configured directly under the Layer 3 dynamic template.

Applying the PPPoE Configurations to an Access Interface

Perform this task to apply the PPPoE profiles and the policy-maps to an access interface. The completion of this task enables the receiving of PPPoE traffic on the interface.

Before you begin

You must perform this task after performing the Creating PPPoE Profiles, on page 93.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **service-policy type control subscriber** *policy_name*
4. **pppoe enable bba-group** *bbagroup_name*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 5.1`	Enters interface configuration mode for the bundle-interface.
Step 3	**service-policy type control subscriber** *policy_name* **Example:** `RP/0/RSP0/CPU0:router(config-if)# service-policy type control subscriber PL1`	Associates a subscriber control service policy to the interface.
Step 4	**pppoe enable bba-group** *bbagroup_name* **Example:** `RP/0/RSP0/CPU0:router(config-if)# pppoe enable bba-group bba_1`	Enables PPPoE on the bundle-ether interface and specifies the PPPoE profile named bba_1 to be used on this interface. **Note** It is not recommended to remove the call flow-initiated configurations, after subscriber sessions are active. Therefore, you must not delete the **pppoe enable** command from the sub-interface, while the PPPoE sessions are up.
Step 5	**commit**	

Applying the PPPoE Configurations to an Access Interface: An example

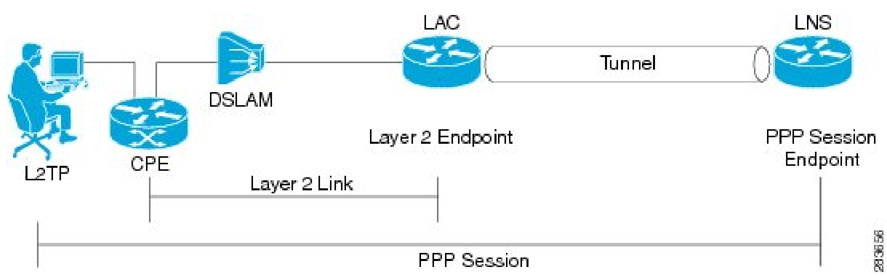
```
configure
interface Bundle-Ether100.10
service-policy type control subscriber PL1
pppoe enable bba-group bba_1
```

```
!
!
end
```

Provisioning PPP LAC Session

In a PPP LAC session, the PPP session is tunneled to a remote network server by BNG, using Layer 2 Tunneling Protocol (L2TP). BNG performs the role of L2TP Access Concentrator (LAC), as it puts the subscriber session in the L2TP tunnel. The device on which the tunnel terminates is called L2TP Network Server (LNS). During a PPP LAC session, the PPPoE encapsulation terminates on BNG; however, the PPP packets travel beyond BNG to LNS through the L2TP tunnel. A typical LAC session is depicted in Figure 1.

Figure 14: LAC Session



The PPP LAC session is used in the Access Network Provider (wholesale) model, where the network service provider (NSP) is a separate entity from the local access network provider (ANP). NSPs perform access authentication, manage and provide IP addresses to subscribers, and are responsible for overall service. The ANP is responsible for providing the last-mile digital connectivity to the customer, and for passing on the subscriber traffic to the NSP. In this kind of setup, the ANP owns the LAC and the NSP owns the LNS.

A PPP LAC session establishes a virtual point-to-point connection between subscriber device and a node in the service provider network. The subscriber dials into a nearby L2TP access connector (LAC). Traffic is then securely forwarded through the tunnel to the LNS, which is present in service provider network. This overall deployment architecture is also known as Virtual Private Dial up Network (VPDN).

The process of provisioning a PPP LAC session involves:

- Defining a template with specific settings for the VPDN. See, Configuring the VPDN Template, on page 100.
- Defining the maximum number of VPDN sessions that can be established simultaneously. See, Configuring Maximum Simultaneous VPDN Sessions, on page 102.
- Activating the logging of VPDN event messages. See, Activating VPDN Logging, on page 103.
- Specifying the method to apply calling station-ID. See, Configuring Options to Apply on Calling Station ID, on page 104.
- Specifying the session-ID. See, Configuring L2TP Session-ID Commands, on page 105.
- Defining specific settings for the L2TP class. See, Configuring L2TP Class Options, on page 105.
- Preventing creation of additional VPDN sessions. See, Configuring Softshut for VPDN, on page 108.

This is a sample user-profile for L2TP LAC:

```
abc_xyz@domain.com Password="abc"
    Service-Type = Outbound-User,
    Tunnel-Type = L2TP,
    Tunnel-Medium-Type = IP,
    Cisco-avpair = "vpdn:ip-addresses=3.3.3.3",
    Cisco-avpair = "vpdn:source-ip=1.1.1.1"
```

> **Note** For L2TP LAC session to be up, the user-profile coming from the RADIUS server to the BNG must have **Service-Type = Outbound-User** configured for the user.

Reassembly of fragmented L2TP data packets is enabled on LAC to prevent these packets from getting dropped. See, L2TP Reassembly on LAC, on page 108

A PPP LAC session supports stateful switchover (SSO) along with non-stop routing (NSR) to reduce traffic loss during RP failover. For more information, see L2TP Access Concentrator Stateful Switchover, on page 110

Restrictions for PPP LAC

- PPPoE LAC is not supported with LC subscribers.
- Provisioning PPP LAC session is subjected to a restriction that only ASR 9000 Enhanced Ethernet Line Cards are supported as core facing line cards.
- A maximum of 19 LNS IP address can be configured in the user-profile for L2TP LAC sessions. This means there can be up to 19 IP addresses assigned to the Tunnel-Server-Endpoint argument for traffic to be securely forwarded through the L2TP tunnel.

  > **Note** If there are more than 19 LNS IP addresses, they are rejected, which means the previous 19 addresses are not overwritten with the new addresses.

Configuring the VPDN Template

Perform this task to configure the vpdn template:

SUMMARY STEPS

1. **configure**
2. **vpdn template**
3. **l2tp-class** *class_name*
4. **tunnel busy timeout** *timeout_value*
5. **caller-id mask-method remove match** *match_substring*
6. **dsl-line-info-forwarding**
7. **ip tos** *type_of_service_value*
8. **vpn id** *value*
9. **vpn vrf** *vrf_name*
10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**vpdn template** **Example:** `RP/0/RSP0/CPU0:router(config)# vpdn template`	Enters the VPDN template sub-mode.
Step 3	**l2tp-class** *class_name* **Example:** `RP/0/RSP0/CPU0:router(config-vpdn-template)# l2tp-class class_temp`	Configures the l2tp class command.
Step 4	**tunnel busy timeout** *timeout_value* **Example:** `RP/0/RSP0/CPU0:router(config-vpdn-template)# tunnel busy timeout 456`	Configure l2tp tunnel busy list commands. The busy timeout value ranges from 60-65535.
Step 5	**caller-id mask-method remove match** *match_substring* **Example:** `RP/0/RSP0/CPU0:router(config-vpdn-template)# caller-id mask-method remove match m1`	Configures options to apply on calling station id by masking the characters by the match substring specified.
Step 6	**dsl-line-info-forwarding** **Example:** `RP/0/RSP0/CPU0:router(config-vpdn-template)# dsl-line-info-forwarding`	Forwards the DSL Line Info attributes.
Step 7	**ip tos** *type_of_service_value* **Example:** `RP/0/RSP0/CPU0:router(config-vpdn-template)# ip tos 56`	Sets IP ToS value for tunneled traffic. The service value ranges from 0 to 255.
Step 8	**vpn id** *value* **Example:** `RP/0/RSP0/CPU0:router(config-vpdn-temp)# vpn id 3333:33`	Specifies tunnel for a vpn and configures the vpn id with the value 3333:33. The value ranges from 0-ffffff in hexadecimal.
Step 9	**vpn vrf** *vrf_name* **Example:** `RP/0/RSP0/CPU0:router(config-vpdn-template)# vpn vrf vrf_1`	Configures the vpn vrf name.

	Command or Action	Purpose
Step 10	**commit**	

Configuring the VPDN Template: An example

```
configure
l2tp-class class hello-interval 100
vpdn
template l2tp-class class //template default will be used and display in show run
template tunnel busy timeout 567
l2tp-class class

vpdn
template default
l2tp-class class
!
end
```

Configuring Maximum Simultaneous VPDN Sessions

Perform this task to configure the maximum simultaneous vpdn sessions for session limiting per tunnel:

SUMMARY STEPS

1. **configure**
2. **vpdn**
3. **session-limit** *number_of_sessions*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**vpdn** **Example:** `RP/0/RSP0/CPU0:router(config)# vpdn`	Enables VPDN and enters the VPDN sub-mode.
Step 3	**session-limit** *number_of_sessions* **Example:** `RP/0/RSP0/CPU0:router(config-vpdn)# session-limit 200`	Configures the maximum simultaneous VPDN sessions. The range is from 1 to 131072. **Note** If limit is configured after a number of sessions are up, then those sessions remain up irrespective of the limit.
Step 4	**commit**	

Configuring Maximum Simultaneous VPDN Sessions: An example

```
configure
vpdn
session-limit 200
!
end
```

Activating VPDN Logging

Perform this task to activate logging of VPDN event information. When VPDN event logging is enabled, VPDN event messages are logged as the events occur.

Note Tunnel start and stop records are generated without any tunnel statistics.

SUMMARY STEPS

1. **configure**
2. **vpdn**
3. **logging** [**cause**| **cause-normal** | **dead-cache** | **local** | **tunnel-drop** | **user**]
4. **history failure**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**vpdn** **Example:** `RP/0/RSP0/CPU0:router(config)# vpdn`	Enters the VPDN sub-mode.

Step 3	**logging** [**cause**| **cause-normal** | **dead-cache** | **local** | **tunnel-drop** | **user**]
	Example:
	`RP/0/RSP0/CPU0:router(config-vpdn)# logging local`
	`RP/0/RSP0/CPU0:router(config-vpdn)# logging user`
	`RP/0/RSP0/CPU0:router(config-vpdn)# logging cause`
	`RP/0/RSP0/CPU0:router(config-vpdn)# logging tunnel-drop`
		Enables the logging of generic VPDN events.

Step 4	**history failure** **Example:** `RP/0/RSP0/CPU0:router(config-vpdn)# history failure`	Enables logging of VPDN failure events to the history failure table.

	Command or Action	Purpose
Step 5	**commit**	

Activating VPDN Logging: An example

```
configure
vpdn
history failure
logging local
logging user
logging cause-normal
logging tunnel-drop
logging dead-cache
!
end
```

Configuring Options to Apply on Calling Station ID

Perform this task to configure options to apply on calling station ID. The calling station ID provides detailed information about the originator of the session, such as the phone number of the originator, the Logical Line ID (LLID) used to make the connection on the LAC, or the MAC address of the PC connecting to the network.

SUMMARY STEPS

1. **configure**
2. **vpdn**
3. **caller-id mask-method remove match** *match_name*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**vpdn** **Example:** `RP/0/RSP0/CPU0:router(config)# vpdn`	Enters the VPDN sub-mode.
Step 3	**caller-id mask-method remove match** *match_name* **Example:** `RP/0/RSP0/CPU0:router(config-vpdn)# caller-id mask-method remove match match_class`	Suppresses the calling station ID for all users. If there is a 'match' option, then calling station ID only for users which have the 'match-string' in their username is suppressed. **Note** This command an also be run under the vpdn template configuration mode.
Step 4	**commit**	

Configuring Options to Apply on Calling Station ID: An example

```
configure
vpdn //or vpdn template
caller-id mask-method remove match match_call
!
end
```

Configuring L2TP Session-ID Commands

Perform this task to configure L2TP session-id commands.

SUMMARY STEPS

1. **configure**
2. **vpdn**
3. **l2tp session-id space hierarchical**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**vpdn** **Example:** `RP/0/RSP0/CPU0:router(config)# vpdn`	Configures vpdn.
Step 3	**l2tp session-id space hierarchical** **Example:** `RP/0/RSP0/CPU0:router(config-vpdn)# l2tp session-id space hierarchical`	Enables the hierarchical session-ID allocation algorithm.
Step 4	**commit**	

Configuring L2TP Session-ID Commands: An example

```
configure
vpdn
l2tp session-id space hierarchical
!
end
```

Configuring L2TP Class Options

Perform this task to configure the various options for L2TP class.

SUMMARY STEPS

1. **configure**
2. **l2tp-class** *class_name*
3. **authentication** [**disable** | **enable**]
4. **congestion control**
5. **digest** [**check disable** |**hash** { **MD5** | **SHA1** } | **secret** { **0** | **7** | **LINE** }]
6. **hello-interval** *interval_duration*
7. **hostname** *host_name*
8. **receive-window** *size*
9. **retransmit initial** [**retries** | *retries_number* | **timeout** {**max** *max_seconds* | **min** *min_seconds* }
10. **timeout** [**no-user**{ *timeout_value* | **never** } | **setup** *setup_value*]
11. **tunnel accounting** *accounting_method_list_name*
12. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**l2tp-class** *class_name* **Example:** `RP/0/RSP0/CPU0:router(config)# l2tp-class class1`	Configures the L2TP class command.
Step 3	**authentication** [**disable** | **enable**] **Example:** `RP/0/RSP0/CPU0:router(config-l2tp-class)# authentication disable`	Enables the tunnel authentication. The Enable and Disable options enables or disables the L2TP tunnel authentication.
Step 4	**congestion control** **Example:** `RP/0/RSP0/CPU0:router(config-l2tp-class)# congestion control`	Enables L2TP congestion control.

Step 5	**digest** [**check disable** |**hash** { **MD5** | **SHA1** } | **secret** { **0** | **7** | **LINE** }]
	Example:
	`RP/0/RSP0/CPU0:router(config-l2tp-class)# digest check disable`
	`RP/0/RSP0/CPU0:router(config-l2tp-class)# digest hash MD5`
	`RP/0/RSP0/CPU0:router(config-l2tp-class)# digest secret 0`
		Messages the Digest configuration for L2TPv3 control connection.

Step 6	**hello-interval** *interval_duration* **Example:**	Sets HELLO message interval for specified amount of seconds.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-l2tp-class)# hello-interval 45`	
Step 7	**hostname** *host_name* **Example:** `RP/0/RSP0/CPU0:router(config-l2tp-class)# hostname local_host`	Sets the local hostname for control connection authentication.
Step 8	**receive-window** *size* **Example:** `RP/0/RSP0/CPU0:router(config-l2tp-class)# receive-window 56`	Receives window size for the control connection. The range is from 1 to 16384.

Step 9	**retransmit initial** [**retries** | *retries_number* | **timeout** {**max** *max_seconds* | **min** *min_seconds* }
	Example:
	`RP/0/RSP0/CPU0:router(config-l2tp-class)# retransmit initial retries 58`
	`RP/0/RSP0/CPU0:router(config-l2tp-class)# retransmit initial timeout max 6`
		Receives window size for the control connection. The range is from 1 to 16384.

Step 10	**timeout** [**no-user**{ *timeout_value* | **never** } | **setup** *setup_value*]
	Example:
	`RP/0/RSP0/CPU0:router(config-l2tp-class)# timeout no-user 56`
	`RP/0/RSP0/CPU0:router(config-l2tp-class)# retransmit setup 60`
		Receives window size for the control connection. The timeout value range, in seconds, is from 0 to 86400. The setup value range is from 60 to 6000.

Step 11	**tunnel accounting** *accounting_method_list_name* **Example:** `RP/0/RSP0/CPU0:router(config-l2tp-class)# tunnel accounting acc_tunn`	Configures the AAA accounting method list name.
Step 12	**commit**	

Configuring L2TP Class Options: An example

```
configure
l2tp-class class1
authentication enable
congestion-control
digest check disable
hello-interval 876
hostname l2tp_host
```

```
receive-window 163
retransmit initial timeout 60
timeout no-user 864
tunnel accounting aaa_l2tp
!
end
```

Configuring Softshut for VPDN

Perform this task to configure softshut for vpdn.

SUMMARY STEPS

1. **configure**
2. **vpdn**
3. **softshut**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**vpdn** **Example:** `RP/0/RSP0/CPU0:router(config)# vpdn`	Enters the VPDN sub-mode.
Step 3	**softshut** **Example:** `RP/0/RSP0/CPU0:router(config-vpdn)# softshut`	Ensures that no new sessions are allowed.
Step 4	**commit**	

Configuring Softshut for VPDN: An example

```
configure
vpdn
softshut
!
end
```

L2TP Reassembly on LAC

The L2TP Reassembly feature on L2TP Access Concentrator (LAC) ensures reassembly of fragmented L2TP data packets in the intervening network, between the LAC and L2TP Network Server (LNS). Data packets are fragmented when they exceed the Maximum Transmission Unit (MTU) of the IPv4 core. Enabling this feature prevents the fragmented packets from getting dropped and ensures the subsequent forwarding of these data packets.

When L2TP Reassembly feature is disabled on LAC, fragmented data packets are dropped. The feature does not affect the reassembly of non-L2TP packets. To ensure that packets for non-L2TP applications are properly reassembled regardless of whether load balancing occurs for each packet, it is recommended that:

- A separate loopback address be configured only for L2TP traffic. No other applications on the router should use this IP address.
- Multiple loopback addresses be used for L2TP, but no other applications across all VRFs should use these addresses.

In cases of reassembly errors or fragmentation timeout, the maximum period a traffic flow is kept, before it is forwarded to the Route Switch Processor (RSP) is 250ms.

Restrictions

Enabling L2TP reassembly feature is subjected to these restrictions:

- Only ASR 9000 Enhanced Ethernet Line Cards support L2TP reassembly feature
- Only IPv4 fragmented packets are reassembled
- Only packets with two fragments are reassembled
- The fragments must not overlap
- The fragmented IP headers must not contain options
- The fragmented L2TP packets must be delivered to the same line card. In other words, the intervening network must not use per packet load balancing schemes that make the fragments arrive on different line cards. On the other hand, the reassembly of non-L2TP packets is not affected even when the packets arrive on different line cards

Enabling L2TP Reassembly on LAC

Perform this task to enable L2TP reassembly on L2TP Access Concentrator (LAC):

SUMMARY STEPS

1. **configure**
2. **vpdn**
3. **l2tp reassembly**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**vpdn** **Example:** `RP/0/RSP0/CPU0:router(config)# vpdn`	Enters the VPDN configuration mode.

	Command or Action	Purpose
Step 3	**l2tp reassembly** **Example:** `RP/0/RSP0/CPU0:router(config-vpdn)# l2tp reassembly`	Enables L2TP reassembly on LAC.
Step 4	**commit**	

Enabling L2TP Reassembly on LAC: An example

```
configure
vpdn
l2tp reassembly
!
end
```

L2TP Access Concentrator Stateful Switchover

The L2TP Access Concentrator Stateful Switchover (LAC SSO) feature establishes one of the RPs as the active processor, designates the other RP as the standby processor, and then synchronizes critical state information between them. In specific Cisco networking devices that support dual RPs, LAC SSO takes advantage of RP redundancy to increase network availability.

LAC SSO supports non-stop routing (NSR) for VPDN and L2TP protocols in the event of a RP failover. The NSR provides the ability to guarantee reliable L2TP and VPDN synchronization between active and standby RPs. In case of RP fail-over, all VPDN and L2TP tunnels and sessions information are preserved without impacting the L2TP network peer. Also, peer networking devices do not experience routing flaps, and therefore reduce loss of service outages for customers. When VPDN and LAC SSO are enabled, all the tunnels and sessions are mirrored to the backup RP.

Enabling LAC SSO

Perform this task to enable LAC/VPDN SSO feature:

SUMMARY STEPS

1. **configure**
2. **vpdn**
3. **redundancy**
4. **commit**
5. **show vpdn redundancy**
6. **show vpdn redundancy mirroring**
7. **show l2tpv2 redundancy**
8. **show l2tpv2 redundancy mirroring**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**vpdn** **Example:** `RP/0/RSP0/CPU0:router(config)# vpdn`	Enters vpdn configuration mode.
Step 3	**redundancy** **Example:** `RP/0/RSP0/CPU0:router(config-vpdn)# redundancy`	Enters vpdn redundancy configuration mode.
Step 4	**commit**	
Step 5	**show vpdn redundancy** **Example:** `RP/0/RSP0/CPU0:router# show vpdn redundancy`	Displays all vpdn redundancy related information.
Step 6	**show vpdn redundancy mirroring** **Example:** `RP/0/RSP0/CPU0:router# show vpdn redundancy mirroring`	Displays vpdn related mirroring statistics.
Step 7	**show l2tpv2 redundancy** **Example:** `RP/0/RSP0/CPU0:router# show l2tpv2 redundancy`	Displays L2TP redundancy related information.
Step 8	**show l2tpv2 redundancy mirroring** **Example:** `RP/0/RSP0/CPU0:router# show l2tpv2 redundancy mirroring`	Displays L2TP related mirroring statistics.

Enabling LAC SSO: Example

```
configure
  vpdn
    redundancy
      process-failures switchover
end
```

Enabling RPFO on Process-failures

In the event of an application or process crash, if VPDN NSR is enabled, an RP failover is triggered and a new primary RP process restarts without traffic loss.

The VPDN NSR is disabled by default. Perform these steps to enable RPFO:

SUMMARY STEPS

1. **configure**
2. **nsr process-failures switchover**
3. **vpdn**
4. **redundancy**
5. **process-failures switchover**
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**nsr process-failures switchover** **Example:** `RP/0/RSP0/CPU0:router(config)# l2tp nsr process-failures switchover`	Enables VPDN non-stop routing.
Step 3	**vpdn** **Example:** `RP/0/RSP0/CPU0:router(config)# vpdn`	Enters vpdn configuration mode.
Step 4	**redundancy** **Example:** `RP/0/RSP0/CPU0:router(config-vpdn)# redundancy`	Enters vpdn redundancy configuration mode.
Step 5	**process-failures switchover** **Example:** `RP/0/RSP0/CPU0:router(config-vpdn-redundancy)# process-failures switchover`	Forces a switchover in case of a process failure.
Step 6	**commit**	

PPPoE Smart Server Selection

The PPPoE Smart Server Selection (PADO delay) feature in BNG allows the PPPoE client to control the selection of BNG for session establishment, in a multi-BNG setup. The feature provides the option for configuring a delay in sending PADO messages from BNG, in response to the PADI messages received from the PPPoE clients. This, in turn, helps in establishing a priority order and load balancing across all BNGs.

When establishing a PPPoE session in a multi-BNG setup, the clients broadcast their PADI messages to all BNGs. When the BNGs reply with a PADO message, the subscriber selects a BNG, and sends a PADR message to the BNG with which a session needs to be established. Most PPPoE clients send a PADR message to the BNG from which it received the first PADO message. By configuring the Smart Server Selection feature on BNG, a delay is added to the PADO messages sent from the BNG, based on the properties of the PADI messages received from the PPPoE clients. This delay in receiving the PADO packets, in turn, gives the PPPoE client the flexibility of effectively selecting the appropriate BNG to which the PADR message is to be sent.

Configuration options for Smart Server Selection

- Allows configuring a specific delay for the PADO message sent from BNG.
- Allows configuring a delay for the PADO message sent from BNG, based on the Circuit-ID, Remote-ID and Service-Name contained in the incoming PADI message.
- Allows Circuit-ID and Remote-ID tag matching, with strings up to 64 characters in length.
- Allows partial matching on Circuit-ID, Remote-ID, and Service-Name contained in the incoming PADI message.

For configuring the delay for a PADO message, see Configuring PADO Delay, on page 113.

Configuring PADO Delay

Perform this task to configure a delay for PPPoE Active Discovery Offer (PADO) message, or in other words, enabling Smart Server Selection feature for a PPPoE BBA-Group in BNG.

Note If multiple delays match a particular subscriber, Circuit-ID matches are preferred to Remote-ID matches, which, in turn, are preferred to Service-Name matches.

SUMMARY STEPS

1. **configure**
2. **pppoe bba-group** *bba-group-name*
3. Use these commands to configure the PADO delay based on a specific delay value, Circuit-ID, Remote-ID, and Service-Name respectively:
   - **pado delay** *delay*
   - **pado delay circuit-id** {*delay* | {**string** | **contains**} *string delay*}
   - **pado delay remote-id** {*delay* | {**string** | **contains**} *string delay*}
   - **pado delay service-name** {**string** | **contains**} *string delay*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**pppoe bba-group** *bba-group-name* **Example:**	Enters the PPPoE BBA-Group configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# pppoe bba-group bba_1`	

Step 3	Use these commands to configure the PADO delay based on a specific delay value, Circuit-ID, Remote-ID, and Service-Name respectively:
	• **pado delay** *delay*
	• **pado delay circuit-id** {*delay* | {**string** | **contains**} *string delay*}
	• **pado delay remote-id** {*delay* | {**string** | **contains**} *string delay*}
	• **pado delay service-name** {**string** | **contains**} *string delay*
	Example:
	`RP/0/RSP0/CPU0:router(config-bbagroup)# pado delay 500`
	`RP/0/RSP0/CPU0:router(config-bbagroup)# pado delay circuit-id 200`
	`RP/0/RSP0/CPU0:router(config-bbagroup)# pado delay remote-id string circuit4`
	`RP/0/RSP0/CPU0:router(config-bbagroup)# pado delay service-name contains service 9950`
		Sets the PADO delay in milliseconds based on:
		• A specific *delay* value
		• Circuit-ID received in PADI
		• Remote-ID received in PADI
		• Service-Name received in PADI
		The *delay* range is from 0 to 10000.
		The **string** option delays the PADO message, when the Circuit-ID (or Remote-ID or Service-Name) received in the PADI message matches the configured *string* value.
		The **contains** option delays the PADO message, when the Circuit-ID (or Remote-ID or Service-Name) received in the PADI message contains the configured *string* value.

Step 4	**commit**	

Configuring PPPoE PADO delay : An example

```
 pppoe bba-group bba_1
 pado delay 500
 pado delay remote-id 100
 pado delay circuit-id string circuit4 8000
 pado delay service-name contains service 9950
!
end
```

PPPoE Session Limit, Throttle and In-flight-window

PPPoE Session Limit

The PPPoE Session Limit support limits the number of PPPoE sessions that can be created on a BNG router. As a result, it reduces excessive memory usage by the BNG router for virtual access.

This offers additional configuration flexibility on the BNG router by limiting the number of PPPoE sessions for each:

- Line card
- Parent interface

- Peer MAC address
- Peer MAC address under individual access interface
- Circuit-ID
- Remote-ID
- Combination of Circuit-ID and Remote ID
- Access interface using the same Inner VLAN tag
- Access interface using the same Outer VLAN tag.
- Access interface using the same Inner and Outer VLAN tags

The PPPoE Session Limit support also limits the number of Inter Working Function (IWF) sessions for each peer MAC address and for each peer MAC address under individual access interface.

See, Configuring PPPoE Session Limit, on page 115.

Configuring PPPoE Session Limit

Perform this task to configure PPPoE session limit for a PPPoE BBA-Group in BNG.

SUMMARY STEPS

1. **configure**
2. **pppoe bba-group** {*bba-group name*}
3. **sessions** {**access-interface** | **circuit-id** | **circuit-id-and-remote-id** | **inner-vlan** | {{**mac** | **mac-iwf**} [**access-interface**] }} | **max** | **outer-vlan** | **remote-id** | **vlan**} **limit** *limit-count* [**threshold** *threshold-count*]
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**pppoe bba-group** {*bba-group name*} **Example:** `RP/0/RSP0/CPU0:router(config)# pppoe bba-group bba_1`	Enters the specific PPPoE BBA-Group configuration mode.
Step 3	**sessions** {**access-interface** | **circuit-id** | **circuit-id-and-remote-id** | **inner-vlan** | {{**mac** | **mac-iwf**} [**access-interface**] }} | **max** | **outer-vlan** | **remote-id** | **vlan**} **limit** *limit-count* [**threshold** *threshold-count*] **Example:**	Configures the PPPoE session limits. If the optional argument, **threshold** is configured, a log message is generated when the PPPoE session limit exceeds the *threshold-count* value. The *limit-count* value and *threshold-count* value ranges from 1 to 65535. The default value is 65535.

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router(config-bbagroup)# sessions access-interface limit 1000`
	`RP/0/RSP0/CPU0:router(config-bbagroup)# sessions mac access-interface limit 5000 threshold 4900`
	`RP/0/RSP0/CPU0:router(config-bbagroup)# sessions circuit-id limit 8000 threshold 7500`	

Step 4	**commit**	

Configuring PPPoE Session Limit: An example

```
configure
pppoe bba-group bba1
 sessions circuit-id limit 8000 threshold 7500
 sessions access-interface limit 1000
 sessions mac access-interface limit 5000 threshold 900
!
end
```

PPPoE Session Throttle

The PPPoE Session Throttle support on BNG limits the number of PPPoE session requests coming to BNG within a specified period of time. This, in turn, ensures that the session establishment of other client requests coming to the BNG server is not impacted.

This offers configuration flexibility in the BNG router by throttling the number of session requests based on one of these:

- Peer MAC address
- Peer MAC address under individual access interface
- Circuit-ID
- Remote-ID
- A combination of Circuit-ID and Remote ID
- Inner VLAN tag under individual access interface
- Outer VLAN tag under individual access interface
- Inner and Outer VLAN tag under individual access interface

The PPPoE session throttle support also throttles the number of Inter Working Function (IWF) session requests for each peer MAC address under an individual access interface.

See, Configuring PPPoE Session Throttle, on page 116.

Configuring PPPoE Session Throttle

Perform this task to configure PPPoE session throttle for a PPPoE BBA-Group in BNG.

SUMMARY STEPS

1. **configure**
2. **pppoe bba-group** *bba-group name*
3. **sessions** {**circuit-id** | **circuit-id-and-remote-id** | **inner-vlan** | {**mac** [**access-interface**] } | {**mac-iwf** {**access-interface**}} | **outer-vlan** | **remote-id** | **vlan**} **throttle** *request-count request-period blocking-period*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**pppoe bba-group** *bba-group name* **Example:** `RP/0/RSP0/CPU0:router(config)# pppoe bba-group bba_1`	Enters the PPPoE BBA-Group configuration mode.

Step 3	**sessions** {**circuit-id** | **circuit-id-and-remote-id** | **inner-vlan** | {**mac** [**access-interface**] } | {**mac-iwf** {**access-interface**}} | **outer-vlan** | **remote-id** | **vlan**} **throttle** *request-count request-period blocking-period*
	Example:
	`RP/0/RSP0/CPU0:router(config-bbagroup)# sessions circuit-id throttle 1000 50 25`
	`RP/0/RSP0/CPU0:router(config-bbagroup)# sessions mac-iwf access-interface throttle 5000 100 50`
		Configures the PPPoE session throttles.
		The *request-count* value ranges from 1 to 65535.
		The *request-period* value ranges from 1 to 100.
		The *blocking-period* value ranges from 1 to 100.

Step 4	**commit**	

Configuring PPPoE Session Throttle: An example

```
configure
pppoe bba-group bba1
 sessions circuit-id throttle 1000 50 25
 sessions mac-iwf access-interface throttle 5000 100 50
!
```

PPPoE In-flight-window

PPPoE in-flight-window is an enhancement to limit the number of PPPoE sessions in BNG that are in progression towards established state. The in-flight-window option sets the PPPoE process queue to a particular limit per LC and per RP, thereby providing a better control of incoming PPPoE sessions to BNG.

To enable this feature, use **pppoe in-flight-window** command in the global configuration mode.

Note The recommended in-flight-window *size* for RP-based subscribers is 200, and that for LC-based subscribers is 50. Values higher than these are not recommended for production deployment, as it can lead to system instability.

Configuration Example for PPPoE In-flight-window

```
Router# configure
Router(config)# pppoe in-flight-window 200
Router(config)#commit
```

Activating IPv6 Router Advertisement on a Subscriber Interface When IPv4 Starts

BNG introduces the ability to automatically trigger an IPv6 router advertisement on an IPv4 subscriber interface. This feature can be used by subscriber interfaces that are on a dual stack network and are enabled for IPv6 processing.

To configure this feature you can either use dynamic templates through CLI or configure RADIUS user profile attributes. This feature is only supported for subscriber sessions that use the IPoE protocol.

In a BNG dual stack network, an IPv4 session is initiated first followed by an IPv6 session request. After receiving the DHCP IPv6 request, the DHCP server allocates an IPv6 address.

Creating Dynamic Template for Enabling IPv6 Router Advertisement on an IPv4 Susbscriber Interface

Perform this task to create a dynamic template to enable IPv6 router advertisements on a subscriber interface:

SUMMARY STEPS

1. **configure**
2. **dynamic-template**
3. **type ipsubscriber** *dynamic template name*
4. **ipv6 nd start-ra-on-ipv6-enable**
5. **show ipv6 nd idb interface** *subscriber interface* **detail location** *member location*

DETAILED STEPS

Step 1 **configure**

Step 2 **dynamic-template**

Enters the dynamic template configuration.

Example:

```
RP/0/RSP0/CPU0:router(config)#dynamic-template
```

Step 3 **type ipsubscriber** *dynamic template name*

Creates a dynamic template with a user-defined name for an ipsubscriber service.

Example:

```
RP/0/RSP0/CPU0:router(config-dynamic-template)#type ipsubscriber ipoe_ipv6
```

Step 4 **ipv6 nd start-ra-on-ipv6-enable**

Enables IPv6 router advertisement capability if ipv6-enable is already configured, instead of waiting for the dual stack to boot up.

Example:

```
RP/0/RSP0/CPU0:router(config-dynamic-template)#type ipsubscriber ipoe_ipv6 start-ra-on-ipv6-enable
```

Step 5 **show ipv6 nd idb interface** *subscriber interface* **detail location** *member location*

Example:

```
RP/0/RSP0/CPU0:router##show ipv6 nd idb interface bundle-ether 50.11.ip3 d
RA flag: 0x1, Unicast RA send: FALSE, Initial RA count: 3, RA pkts sent count: 4
```

Displays the RA packets sent from the subscriber interface.

Making DHCP Settings

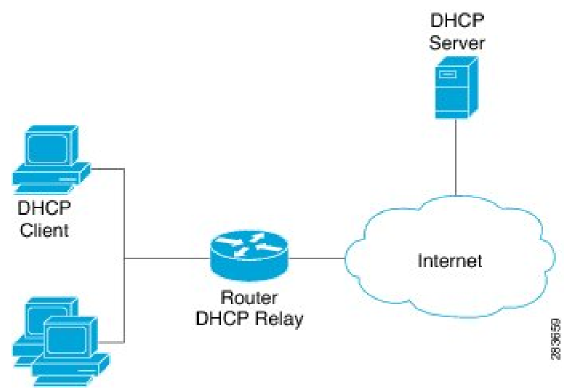
Note For detailed information on the DHCP features and configurations supported on ASR9K router, refer to the *Implementing the Dynamic Host Configuration Protocol* chapter in the *IP Addresses and Services Configuration Guide for Cisco ASR 9000 Series Routers*. For a complete list of DHCP commands supported on ASR9K router, refer to the *DHCP Commands* chapter in the *IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers*.

The Dynamic Host Configuration Protocol (DHCP) is a network protocol used to configure network devices so that they can communicate on an IP network. There are three distinct elements in a DHCP network:

- DHCP client—It is the device that seeks IP configuration information, such as IP address.
- DHCP server—It allocates IP address from its address pool to the DHCP client.
- DHCP relay or DHCP proxy—It passes IP configuration information between the client and server. It is used when DHCP client and DHCP server are present on different networks.

Initially, the DHCP client (which is a CPE) does not possess an IP address. As a result, it sends a L2 broadcast request to get an IP address. Acting as the relay agent, BNG processes the request and forwards it to the DHCP server. BNG also forwards responses from the DHCP server back to the DHCP client, ensuring that the end device gets correct IP configuration information. A typical DHCP layout is depicted in this figure.

Figure 15: DHCP Network



The DHCP server allocates IP addresses for only a configurable period of time known as the lease period. If a client device needs to retain the IP address for a period longer than the lease period, then the client must renew the lease before it expires. To renew the lease, the client sends a unicast request to the DHCP server. On receiving the request message, the server responds with an acknowledgment, and the client's lease is extended by the lease time specified in the acknowledgment message.

When a control policy is applied to an access interface, it becomes a subscriber access interface. Otherwise, it is a DHCP standalone interface. For the standalone interface, DHCP adds routes to RIB and populates ARP entries, based on the configuration.

For the subscriber access interface, DHCP uses the policy-plane to determine whether the IP subscriber session should be created for a client binding. This is determined based on whether a valid control policy is applied to the access-interface on which the client binding is created. If a subscriber session is created, then a route is added for the subscriber interface, but no ARP requests are sent out from that subscriber interface.

BNG can be configured to either act as DHCP proxy or DHCP server in the DHCP network.

Note DHCP relay is not supported for BNG.

Enabling DHCP Proxy

As the DHCP proxy, BNG performs all the functions of a relay and also provides some additional functions. In the proxy mode, BNG conceals DHCP server details from DHCP clients. BNG modifies the DHCP replies such that the client considers the proxy to be the server. In this state the client interacts with BNG as if it is the DHCP server.

BNG procures IP leases from the DHCP server and keeps it in its pool. When the client needs to renew its lease, it unicasts the lease renewal request directly to the BNG, assuming it to be the server. BNG renews the lease by allocating the lease from its lease pool.

This way the DHCP proxy splits the lease management process into two phases:

- Server to Proxy (Proxy Lease)
- Proxy to Client (Client lease)

The two phase lease management has these features:

- Shorter client lease times and longer proxy lease times.
- High frequency lease management (renews) at network edge.
- Low frequency lease management (renews) at centralized server.

The benefits of DHCP proxy are:

- Reduced traffic between BNG and DHCP server.
- Quicker client response to network outages.

Configuring DHCP proxy on BNG involves these phases:

- Creating a proxy profile. The profile contains various proxy settings. These settings are applied when the profile is attached to an interface. To create a proxy profile, see Configuring DHCP IPv4 Profile Proxy Class, on page 121
    - Specifying client lease period. The client should renew the lease before the completion of this time period, otherwise the lease expires. To specify the client lease period within a proxy profile, see Configuring the Client Lease Time, on page 124.
    - Specifying remote-ID. The remote-ID is used by the proxy to identify the host that had sent the DHCP request. To define a remote-id within a proxy profile, see Configuring a Remote-ID, on page 123.
- Specifying circuit-ID for an interface. The circuit-ID is used by the proxy to identify the circuit in which the DHCP request was received. Later, DHCP proxy uses it for relaying DHCP responses back to the proper circuit. The circuit-ID is defined for an interface. To define it, see Configuring a Circuit-ID for an Interface, on page 122.
- Attaching proxy profile to an interface. See, Attaching a Proxy Profile to an Interface, on page 125

Configuring DHCP IPv4 Profile Proxy Class

Perform this task to define DHCP.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **class** *class-name*
5. **commit**
6. **show dhcp ipv4 proxy profile name** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:**	Enters the IPv4 DHCP configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	
Step 3	**profile** *profile-name* **proxy** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 proxy`	Enters the proxy profile configuration mode. The DHCP Proxy makes use of the class information to select a subset of parameters in a given profile.
Step 4	**class** *class-name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-profile)# class blue`	Creates a DHCP proxy profile class and enters the proxy profile class mode.
Step 5	**commit**	
Step 6	**show dhcp ipv4 proxy profile name** *name* **Example:** `RP/0/RSP0/CPU0:routershow dhcp ipv4 proxy profile name profile1`	(Optional) Displays the details proxy profile information.

Configuring a Circuit-ID for an Interface

Perform this task to configure a circuit-id for an interface.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **interface** *type interface-path-id*
4. **proxy information option format-type circuit-id** *value*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters DHCP IPv4 configuration submode.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# interface Bundle-Ether 355`	Configures the interface and enters the interface configuration mode.
Step 4	**proxy information option format-type circuit-id** *value* **Example:**	Configures the circuit-id for this interface.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-dhcpv4)# proxy information option format-type circuit-id 7`	
Step 5	**commit**	

Configuring a Circuit-ID for an Interface: An example

```
configure
dhcp ipv4
interface Bundle-Ether100.10
proxy information option format-type circuit-id 7
 !
!
end
```

Configuring a Remote-ID

Perform this task to configure a remote-ID.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **relay information option remote-id** *value*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters the IPv4 DHCP configuration mode.
Step 3	**profile** *profile-name* **proxy** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 proxy`	Creates a DHCP proxy profile.
Step 4	**relay information option remote-id** *value* **Example:** `RP/0/RSP0/CPU0:router(config-if)# relay information option remote-id 9`	Inserts relay agent information for remote id suboptions such as remote-ID value.
Step 5	**commit**	

Configuring a Remote-ID: An example

```
configure
dhcp ipv4
profile profile1 proxy
relay information option remote-id 9
 !
!
end
```

Configuring the Client Lease Time

Perform this task to configure the client lease time. It defines the time period after which the client lease expires.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **lease proxy client-lease-time** *value*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters the IPv4 DHCP configuration mode.
Step 3	**profile** *profile-name* **proxy** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 proxy`	Creates a DHCP profile.
Step 4	**lease proxy client-lease-time** *value* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-proxy-profile)# lease proxy client-lease-time 600`	Configures a client lease time for each profile. The minimum value of the lease proxy client time is 300 seconds.
Step 5	**commit**	

Configuring the Client Lease Time: An example

```
configure
dhcp ipv4
profile profile1 proxy
lease proxy client-lease-time 600
```

```
 !
!
end
```

Attaching a Proxy Profile to an Interface

Preform this task to attach a proxy profile to an interface. After it is attached, the various settings specified in the proxy profile take effect on the interface.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **interface** *type interface-path-id* **proxy profile** *profile-name*
4. **commit**
5. **show dhcp ipv4 proxy profile name** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters the IPv4 DHCP configuration mode.
Step 3	**interface** *type interface-path-id* **proxy profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# interface Bundle-Ether 344 proxy profile profile1`	Enters the Interface configuration mode and assigns a proxy profile to an interface.
Step 4	**commit**	
Step 5	**show dhcp ipv4 proxy profile name** *name* **Example:** `RP/0/RSP0/CPU0:router# show dhcp ipv4 proxy profile name profile1`	(Optional) Displays the details proxy profile information.

Attaching a Proxy Profile to an Interface: An example

```
configure
dhcp ipv4
interface Bundle-Ether100.10 proxy profile profile1
proxy information option format-type circuit-id 7
 !
!
end
```

DHCPv4 Server

DHCP server accepts address assignment requests and renewals and assigns the IP addresses from predefined groups of addresses contained within Distributed Address Pools (DAPS). DHCP server can also be configured to supply additional information to the requesting client such as the IP address of the DNS server, the default router, and other configuration parameters. DHCP server can accept broadcasts from locally attached LAN segments or from DHCP requests that have been forwarded by other DHCP relay agents within the network.

The DHCP proxy performs all the functions of a relay and also provides some additional functions. The DHCP proxy conceals DHCP server details from DHCP clients. The DHCP proxy modifies the DHCP replies such that the client considers the proxy to be the server. In this state, the client interacts with the proxy as if it is the DHCP server.

The pool is configured under server-profile-mode and server-profile-class-sub-mode. The class-based pool selection is always given priority over profile pool selection.

Enabling DHCP Server

BNG can be configured to act as a DHCPv4 Server. To create a DHCPv4 Server profile, see Configuring DHCPv4 Server Profile, on page 126.

For more information on DHCPv4 Server configuration, see *Implementing the Dynamic Host Configuration Protocol* chapter in the *IP Addresses and Services Configuration Guide for Cisco ASR 9000 Series Routers*.

Configuring DHCPv4 Server Profile

Perform this task to configure the DHCPv4 Server.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **server**
4. **bootfile** *boot-file-name*
5. **broadcast-flag policy** *unicast-always*
6. **class** *class-name*
7. **exit**
8. **default-router** *address1 address2 ... address8*
9. **lease** { **infinite** | *days minutes seconds* }
10. **limit lease** { **per-circuit-id** | **per-interface** | **per-remote-id** } *value*
11. **netbios-name server** *address1 address2 ... address8*
12. **netbios-node-type** { **number** | **b-node** | **h-node** | **m-node** | **p-node** }
13. **option** *option-code* { **ascii** *string* | **hex** *string* | **ip** *address* }
14. **pool** *pool-name*
15. **requested-ip-address-check disable**
16. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**dhcp ipv4**
	Example:
	`RP/0/RSP0/CPU0:router(config) # dhcp ipv4`
	`RP/0/RSP0/CPU0:router(config-dhcpv4)#`
		Enables DHCP for IPv4 and enters DHCP IPv4 configuration mode.

Step 3	**profile** *profile-name* **server**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4 )# profile TEST server`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
		Enters the server profile configuration mode.

Step 4	**bootfile** *boot-file-name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# bootfile b1`	Configures the boot file.
Step 5	**broadcast-flag policy** *unicast-always* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# broadcast-flag policy unicast-always`	Configures the broadcast-flag policy to unicast-always.

Step 6	**class** *class-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# class Class_A`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile-class)`
		Creates and enters server profile class configuration submode.

Step 7	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile-class)#`	Exits the server profile class submode.

	Command or Action	Purpose
	`exit` `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`	
Step 8	**default-router** *address1 address2 ... address8* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# default-router 10.20.1.2`	Configures the name of the default-router or the IP address.
Step 9	**lease** { **infinite** | *days minutes seconds* } **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# lease infinite`	Configures the lease for an IP address assigned from the pool.
Step 10	**limit lease** { **per-circuit-id** | **per-interface** | **per-remote-id** } *value* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# limit lease per-circuit-id 23`	Configures the limit on a lease per-circuit-id, per-interface, or per-remote-id.
Step 11	**netbios-name server** *address1 address2 ... address8* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# netbios-name-server 10.20.3.5`	Configures the NetBIOS name servers.
Step 12	**netbios-node-type** { **number** | **b-node** | **h-node** | **m-node** | **p-node** } **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# netbios-node-type p-node`	Configures the type of NetBIOS node.
Step 13	**option** *option-code* { **ascii** *string* | **hex** *string* | **ip** *address* } **Example:**	Configures the DHCP option code.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# option 23 ip 10.20.34.56`	
Step 14	**pool** *pool-name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# pool pool1`	Configures the Distributed Address Pool Service (DAPS) pool name.
Step 15	**requested-ip-address-check disable** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# requested-ip-address-check disable`	Validates a requested IP address.
Step 16	**commit**	

Specifying DHCP Lease Limit

The DHCP lease limit feature allows you to limit the number of DHCP bindings on an interface. A binding represents the mapping between the MAC address of the client and the IP address allocated to it. The lease limit can be specified for each Circuit-ID, or Remote-ID, or interface.

The lease limit can be configured through a DHCP proxy profile. When this profile is attached to an interface, bindings up to the configured limit on that interface are allowed. For example, if a profile with a per-circuit lease limit of 10 bindings is assigned to four interfaces, then for each unique Circuit-ID, there would be 10 bindings allowed for each interface.

If the lease limit is lowered below the current number of existing bindings, then the existing bindings are allowed to persist, but no new bindings are allowed to be created until the number of bindings drops below the new lease limit.

If the lease limit is specified from the AAA server, as part of Change of Authorization (CoA) or Access-Accept message, then the DHCP lease limit configured through the proxy profile is overridden. In this case, the most recent session limit, received from the AAA server, is taken as the current lease limit for the particular Circuit-ID. The lease limit set from the AAA server is cleared when there are no more client bindings associated with the Circuit-ID for which the lease limit is applied.

To specify the lease limit, see these procedures:

- Specifying the Lease Limit for a Circuit-ID, on page 130
- Specifying the Lease Limit for a Remote-ID, on page 130
- Specifying the Lease Limit for an Interface, on page 131

Specifying the Lease Limit for a Circuit-ID

Perform this task to specify the lease limit for each Circuit-ID.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **limit lease per-circuit-id** *value*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters the IPv4 DHCP configuration mode.
Step 3	**profile** *profile-name* **proxy** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 proxy`	Creates a DHCP profile.
Step 4	**limit lease per-circuit-id** *value* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-proxy-profile)# limit lease per-circuit-id 1000`	Specifies the lease limit for a Circuit-ID that is applied to an interface.
Step 5	**commit**	

Specifying the Lease Limit for a Circuit-ID: An example

```
configure
dhcp ipv4
profile profile1 proxy
limit lease per-circuit-id 1000
 !
!
end
```

Specifying the Lease Limit for a Remote-ID

Perform this task to specify the lease limit for each Remote-ID.

SUMMARY STEPS

1. **configure**

2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **limit lease per-remote-id** *value*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters the IPv4 DHCP configuration mode.
Step 3	**profile** *profile-name* **proxy** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 proxy`	Creates a DHCP profile.
Step 4	**limit lease per-remote-id** *value* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-proxy-profile)# limit lease per-remote-id 1340`	Specifies the lease limit for a Remote-ID that is applied to an interface.
Step 5	**commit**	

Specifying the Lease Limit for a Remote-ID: An example

```
configure
dhcp ipv4
profile profile1 proxy
limit lease per-remote-id 1340
 !
!
end
```

Specifying the Lease Limit for an Interface

Perform this task to specify the lease limit for each interface.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **limit lease per-interface** *value*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters the IPv4 DHCP configuration mode.
Step 3	**profile** *profile-name* **proxy** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 proxy`	Creates a DHCP profile.
Step 4	**limit lease per-interface** *value* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-proxy-profile)# limit lease per-interface 2400`	Specifies the lease limit for each interface.
Step 5	**commit**	

Specifying the Lease Limit for an Interface: An example

```
configure
dhcp ipv4
profile profile1 proxy
limit lease per-interface 2400
 !
!
end
```

Understanding DHCP Option-82

DHCP Option 82 allows the DHCP server to generate IP addresses based on the location of the client device. This option defines these sub-options:

- Agent Circuit ID Sub-option—This sub-option is inserted by DSLAM and identifies the subscriber line in the DSLAM.
- Agent Remote ID Sub-option—This sub-option is inserted by DSLAM or BNG in an l2-connected topology. It is the client MAC address, but can be overridden. With the DHCP proxy or relay, the client MAC address is lost by the time the packet gets to the DHCP server. This is a mechanism that preserves the client MAC when the packet gets to the server.
- VPN identifier sub-option—This sub-option is used by the relay agent to communicate the VPN for every DHCP request that is sent to the DHCP server, and it is also used to forward any DHCP reply that the DHCP server sends back to the relay agent.
- Subnet Selection Sub-option—This sub-option allows the separation of the subnet from the IP address and is used to communicate with the relay agent. In a DHCP processing, the gateway address specifies

both the subnet on which a DHCP client resides, and the IP address that the server uses to communicate with the relay agent.

- Server Identifier Override Sub-option—This sub-option value is copied in the reply packet from the DHCP server, instead of the normal server ID address. This sub-option contains the incoming interface IP address, which is the IP address on the relay agent that is accessible from the client. Using this information, the DHCP client sends all renew and release packets to the relay agent, which in turn adds all of the VPN sub-options and forwards the renew and release packets to the original DHCP server.

Note The VPN Identifier, Subnet Selection, and Server Identifier Override sub-options are used by DHCP relay/proxy for supporting MPLS VPNs.

Option 82 Relay Information Encapsulation

When two relay agents are relaying messages between the DHCP client and DHCP server, the second relay agent (closer to the server), by default, replaces the first option 82 information with its own option 82. The remote ID and circuit ID information from the first relay agent is lost. In some deployment scenarios, it is necessary to maintain the initial option 82 from the first relay agent, in addition to the option 82 from the second relay agent.

The DHCP option 82 relay information encapsulation feature allows the second relay agent to encapsulate option 82 information in a received message from the first relay agent, if it is also configured to add its own option 82 information. This configuration allows the DHCP server to use option 82 information from both the relay agents.

Configuring DHCPv4 Class of Service (CoS)

BNG supports manual reset of Class of Service (CoS) value of DHCPv4 control packets sent on subscriber interfaces. By default, the outer and inner CoS values are set to 6. This feature allows to set or modify these CoS values sent by BNG.

The inner and outer Class of Service (CoS) values can be configured for DHCPv4 control packets. For broadcast packets, both the **inner-cos** and **outer-cos** commands can be used to configure CoS values. For unicast packets, the **inner-cos** command cannot be directly used. The outer CoS value configured using the **outer-cos** command is also set as the inner CoS value.

To reset the CoS values, use the **dhcp ipv4** [**inner-cos** | **outer-cos**] *value* command.

For more information about configuring the CoS values, see the *BNG DHCP Commands* chapter in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*.

DHCP RADIUS Proxy

BNG 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. For DHCP RADIUS proxy to work, you must configure the DHCPv4 server profile on the BNG interface.

These are the steps involved in the address assignment mechanism:

- The DHCP server sends DHCP client information to the RADIUS server.

- The RADIUS server returns all required information, primarily IPV4 address and subnet mask, to the DHCP server, in the form of RADIUS attributes.
- The DHCP server translates the RADIUS attributes into DHCP options and sends this information back in a DHCP OFFER message to the DHCP client.
- The DHCP binding is synchronized after the RADIUS server authorizes the client session.

If DHCPv4 IETF attributes: **Framed-IP-Address** and **Framed-IP-Netmask** are received from the RADIUS server, then they are preferred and used instead of allocating the IP address from the local pool.

Example:

```
Framed-IP-Address = 10.10.10.81,
Framed-IP-Netmask = 255.255.255.0,
```

If Cisco attribute: **VRF-ID** is received from the RADIUS server and configured on BNG, then it is used and preferred over local configuration.

Example:

```
Cisco-avpair = "vrf-id=RED"
```

If Cisco attributes: **ipv6:addrv6** and **delegated-prefix** are received from the RADIUS server, then they are preferred and used instead of allocating the IP address from the local pool.

Example:

```
Cisco-avpair = "ipv6:addrv6=2000:4:4::1",
Cisco-avpair = "delegated-prefix=3405:100:1015:2::/64"
```

Apart from these attributes, 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 instead of Class A, Class B is used to send the options back to the DHCP client.

Additional RADIUS server attributes are allowed, but not mandatory. The DHCP server ignores additional attributes that it does not recognize. If a RADIUS server user profile contains a required attribute that is empty, the DHCP server does not generate the DHCP options.

Subscriber Session-Restart

BNG supports IPoE subscriber session-restart, where the DHCP binding for a subscriber session is retained even after the session is deleted. The DHCP client still holds the initial IP address issued by BNG. Later, when the client sends data packets or a DHCP renew request, the session is re-created in BNG. This behavior applies to DHCPv4 sessions on RP or LC.

At the time of session deletion, the DHCP binding moves from the BOUND to the DISCONNECT state. The subscriber label is reset to 0x0 when the binding moves to the DISCONNECT state. Later, when the session is re-created, the binding state then moves back from the DISCONNECT to the BOUND. This re-created session has a new subscriber label and a new subscriber interface.

The binding stays in the DISCONNECT state, only till the lease time. If a data packet or renew request does not come before the lease time expires, then the session is cleared.

Session-restart behavior is applicable to session deletions triggered by idle timeout, or by an account-logoff procedure, where the trigger for deletion is any action other than the DHCP release from the client.

Session-restart is not applicable to session deletions done by the execution of the **clear subscriber session all** command. The DHCP bindings are removed in such cases.

For session deletion triggered by the DHCP client, both the session and the DHCP binding are deleted.

Note For session-restart to work, you must configure dual initiators (**initiator dhcp** and **initiator unclassified-source**) under the access-interface.

DHCP Session MAC Throttle

The ASR9K router supports the DHCP session MAC throttle feature. This feature limits the number of DHCP client requests reaching the ASR9K, based on the MAC address of the DHCP clients. This feature is supported for the DHCPv4 proxy, the DHCPv4 server, and the DHCPV6 proxy. The feature prevents a DHCP client from sending multiple DISCOVER packets to the ASR9K router, within short periods of time. This, in turn, prevents that client from impacting the session establishment of other DHCP clients.

A unique throttle entry is created in the system for each unique MAC address received on any interface where the profile is attached.

To configure the DHCP session MAC throttle feature, use the **sessions mac throttle** command in the respective DHCP profile configuration mode.

Configuring DHCP Session MAC Throttle: Example

```
dhcp ipv4
 profile p1 server
  sessions mac throttle 300 60 40
 !
 interface GigabitEthernet0/0/0/0 server profile p1
!
```

DHCPv6 Overview

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) enables DHCP servers to pass configuration parameters, such as IPv6 network addresses, to IPv6 nodes. It enables automatic allocation of reusable network addresses to the requesting clients, using the stateful address-configuration. Along with address and prefix allocation, DHCPv6 also offers additional configuration flexibility by assigning other configuration parameters such as DNS address, DNS domain name, AFTR address to IPv6 nodes in a network.

The basic DHCPv6 client-server concept is similar to using DHCP for IPv4 (DHCPv4). If a client wishes to receive configuration parameters, it sends out a request on the attached local network to detect the available DHCPv6 servers. Although DHCPv6 assigns IPv6 addresses or prefixes, name servers, and other configuration information very similar to that of DHCP for IPv4, these are certain key differences between DHCPv4 and DHCPv6. For example, unlike DHCPv4, address allocation in DHCPv6 is handled using a message option, DHCPv6 clients can request multiple addresses and prefixes in a single request, and DHCPv6 can request different lease times for the addresses and prefixes. These significant advantages of DHCPv6 make it a preferred protocol for address assignment.

IPv6 hosts use Stateless Address Auto-Configuration (SLAAC), a model in which the hosts generate their own addresses using a combination of local and router-advertised information.

The DHCPv6 has been standardized by the IETF through RFC 3315. This DHCPv6 protocol is a stateful counterpart to IPv6 Stateless Address Auto-Configuration (RFC 4862), and can be used separately, or concurrently with SLAAC, to obtain configuration parameters.

Note Prior to configuring DHCPv6, IPv6 must be enabled on the interface on which DHCPv6 is servicing and enable Neighbor Discovery (ND).

For more information about Neighbor Discovery (ND), refer to the "Implementing Network Stack IPv4 and IPv6" section in the *Cisco IOS XR IP Addresses and Services Configuration Guide*.

Restrictions

- DHCPv6 Proxy supports to a maximum of eight external DHCPv6 servers per proxy profile.
- Bulk lease query is not supported.
- DHCPv6 server is supported only with BNG configuration.

DHCPv6 Server and DHCPv6 Proxy

The DHCPv6 server always uses stateful address assignment. On receiving a valid request, the DHCPv6 server assigns IPv6 address or prefix and other configuration attributes such as domain name, domain name server (DNS) address to requesting clients.

A DHCPv6 Relay or Proxy forwards a DHCPv6 message from a client to a server. A DHCPv6 Relay can use either stateless or stateful address assignment. The DHCPv6 Stateless Relay agent acts as an intermediary to deliver DHCPv6 messages between clients and servers. The Relay does not store or keep track of information such as client addresses or the lease time. The DHCPv6 Relay is also known as a Stateless Relay. On the other hand, the DHCPv6 Stateful Relay agent, also known as DHCP proxy, not only forwards a DHCPv6 message from a client to the server, but also keeps track of the client's addresses and lease time. Hence, DHCPv6 Proxy is also known as Stateful Relay. DHCPv6 supports a standalone proxy.

DHCPv6 Proxy enables inserting remote-ID and interface-ID options. The DHCPv6 Proxy uses the interface-ID in addition to remote-ID to choose the interface on which to send the response towards client.

DHCPv6 can be enabled on different configuration modes. For more information about configuring DHCPv6 on different configuring modes, see Enabling DHCPv6 for Different Configuration Modes, on page 136. For more information about setting the DHCPv6 parameters, see Setting Up DHCPv6 Parameters, on page 139.

Note DHCP relay is not supported for BNG.

Enabling DHCPv6 for Different Configuration Modes

Perform this task to enable DHCPv6 for different configuration modes such as global, server profile, proxy profile configuration modes, and server profile class and proxy profile class sub-configuration modes.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv6**
3. **profile** *server_profile_name* **server**
4. **class** *class-name*
5. **dns-server** *address*
6. **domain-name** *name*
7. **prefix-pool** *pool_name*
8. **address-pool** *pool_name*
9. **commit**
10. **interface** *type* *interface-path-id* **server profile** *profile_name*
11. **profile** *proxy_profile_name* **proxy**
12. **link-address** *ipv6_address*
13. **class** *class-name*
14. **helper-address** **vrf** *vrf_name* *ipv6_address*
15. **commit**
16. **interface** *type* *interface-path-id* **proxy profile** *profile_name*
17. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv6** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	Configures DHCP for IPv6 and enters the DHCPv6 configuration mode.
Step 3	**profile** *server_profile_name* **server** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# profile my-server-profile server`	Creates a DHCPv6 server profile and enters the DHCPv6 server profile sub-configuration mode.
Step 4	**class** *class-name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# class server-green`	Defines a class in a server profile and enters the server profile class sub-mode.
Step 5	**dns-server** *address* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# dns-server 1111::1`	Defines a dns-server and the corresponding address in a server profile.
Step 6	**domain-name** *name* **Example:**	Defines a domain name in a server profile.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# domain-name www.xyz.com`	
Step 7	**prefix-pool** *pool_name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# prefix_pool p1`	Configures a prefix pool in a server profile.
Step 8	**address-pool** *pool_name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# address_pool p1`	Configures an address pool in a server profile.
Step 9	**commit**	
Step 10	**interface** *type* *interface-path-id* **server profile** *profile_name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# interface Bundle-Ether1.1 server profile my-server-profile`	Associates a DHCPv6 server configuration profile with an IPv6 interface.
Step 11	**profile** *proxy_profile_name* **proxy** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# profile my-proxy-profile proxy`	Creates a DHCPv6 profile proxy and enters the DHCPv6 proxy sub-configuration mode.
Step 12	**link-address** *ipv6_address* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# link-address 5:6::78`	Specifies the IPv6 address to be filled in the link-address field of the Relay Forward message.
Step 13	**class** *class-name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-proxy-profile)# class proxy-red`	Defines a class in a proxy profile and enters the proxy profile class sub-mode.
Step 14	**helper-address** **vrf** *vrf_name* *ipv6_address* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-proxy-profile)# helper-address vrf my-server-vrf 1:1:1::1`	Configures DHCPv6 address as a helper address to the proxy. **Note** The helper address can be configured only under the proxy profile and proxy profile class sub-modes.
Step 15	**commit**	
Step 16	**interface** *type* *interface-path-id* **proxy profile** *profile_name* **Example:**	Associates a DHCPv6 proxy configuration profile to an IPv6 interface.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-dhcpv6)# interface BundleEther100.1 proxy profile my-proxy-profile`	
Step 17	**commit**	

Enabling DHCPv6 for Different Configuration Modes: An example

```
configure
dhcp ipv6
profile my-server-profile server
link-address 5:6::78
class server-green
dns-server 1111::1
domain-name www.cisco.com
prefix-pool POOL_P6_2
address-pool POOL_A6_1

end
!!
configure
dhcp ipv6
interface GigabitEthernet 0/2/0/0 server profile my-server-profile
profile my-proxy-profile proxy
link-address 5:6::78
class proxy-red
helper-address 5661:11
end
!!
configure
dhcp ipv6
interface GigabitEthernet 0/2/0/0 proxy profile my-proxy-profile
end
!!
```

Setting Up DHCPv6 Parameters

Perform this task to set up DHCPv6 parameters such as address pool name, prefix pool name, DNS server, domain name, lease time, and helper address.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv6**
3. **profile** *server_profile_name* **server**
4. **dns-server** *ipv6_address*
5. **domain-name** *domain_name*
6. **lease**
7. **helper-address** **vrf** *vrf_name* *ipv6_address*
8. **prefix-pool** *prefix-pool-name*
9. **address-pool** *address-pool-name*
10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv6** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	Configures DHCP for IPv6 and enters the DHCPv6 configuration mode.
Step 3	**profile** *server_profile_name* **server** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# profile my-server-profile server`	Configures DHCPv6 server profile and enters the DHCPv6 server profile sub-configuration mode.
Step 4	**dns-server** *ipv6_address* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# dns-server 1:1:1::1`	Configures the DNS server for DHCPv6 server profile. **Note** The DNS server name is defined in the class mode. If the same parameters are defined in the profile mode too, then the values defined in the class mode takes precedence.
Step 5	**domain-name** *domain_name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# domain-name my.domain.name`	Configures the DNS domain name for DHCPv6 server profile. **Note** The DNS server name is defined in the class mode. If the same parameters are defined in the profile mode too, then the values defined in the class mode takes precedence.
Step 6	**lease** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# lease 1 6 0`	Configures the lease time for a duration of 1 day, 6 hours, and 0 minutes.
Step 7	**helper-address vrf** *vrf_name ipv6_address* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-proxy-profile)# helper-address vrf my-server-vrf 1:1:1::1`	Configures DHCPv6 address as a helper address to the proxy. **Note** The helper address can be configured only under the proxy profile and proxy profile class sub-modes.
Step 8	**prefix-pool** *prefix-pool-name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile-class)# prefix-pool my-server-delegated-prefix-pool`	Configures the prefix pool under the DHCPv6 server profile class sub-mode.
Step 9	**address-pool** *address-pool-name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile-class)# address-pool my-server-address-pool`	Configures the address pool under the DHCPv6 server profile class sub-mode.

	Command or Action	Purpose
Step 10	**commit**	

Setting Up DHCPv6 Parameters: An example

```
configure
dhcp ipv6
profile my-server-profile server
dns-server 1:1:1::1
domain-name my.domain.name
lease 1 6 0
class class1
prefix-pool my-server-delegated-prefix-pool
address-pool my-server-address-pool
end
!!
```

DHCPv6 Features

DHCPv6 is widely used in LAN environments to dynamically assign host IP addresses from a centralized server. This dynamic assignment of addresses reduces the overhead of administration of IP addresses. DHCPv6 also helps conserve the limited IP address space. This is because IP addresses no longer need to be permanently assigned to hosts; only those hosts that are connected to the network consume IP addresses.

The DHCPv6 features supported in BNG are:

High Availability Support for DHCPv6

High availability support for DHCPv6 includes:

Linecard Online Insertion and Removal

Linecard Online Insertion and Removal (OIR) enables you to replace faulty parts without affecting the system's operations. When a card is inserted, power is available on the card, and it initializes itself to start being operational.

Note DHCPv6 bindings are not affected by Linecard OIR.

Checkpoint and Shadow Database

The checkpoint and shadow database are actively maintained on the RSP and contains a copy of all bindings from all linecards. The checkpoint database has client or subscriber bindings from the subscribers over interfaces in its scope. The shadow database on the active RSP updates the standby shadow database.

DHCPv6 Hot Standby

DHCPv6 Hot Standby is a process that is supported only on RSPs. Whenever the active RSP stops responding, it is instantly replaced by a standby RSP. The standby RSP takes over processing when it becomes active.

DHCPv6 Prefix Delegation

The DHCPv6 prefix delegation is a mechanism of delegating IPv6 prefixes to a client. The prefix delegation feature can be used to manage link, subnet, and site addressing changes.

An Internet Service Provider (ISP) assigns prefix to a customer for use within the customer's network. Prefix delegation occurs between a provider edge (PE) device and customer premises equipment (CPE), using the DHCPv6 prefix delegation option. After the ISP has delegated prefixes to a customer, the customer may further subnet and assign prefixes to the links in the customer's network.

By default, the prefix delegation feature is always enabled.

IPv6 IPoE Subscriber Support

An IPv6 subscriber transmits IPv6 address that is created using the DHCPv6 protocol. The IPv6 subscribers run IPv6 on the CPE device and are connected to BNG through a Layer-2 network or through Layer-2 aggregation. The IPv6 subscribers are supported when they are directly connected to the BNG or through a Layer-2 aggregator.

To enable IPv6 IPoE subscriber support, the DHCPv6 profile needs to be explicitly configured on the subscriber interface. For more information, see Configuring IPv6 IPoE Subscriber Interface, on page 142.

FSOL Handling

The DHCPv6 First Sign of Life (FSOL) handling is only supported for IPoE sessions. DHCPv6 handles SOLICIT packet from client as FSOL packet for IPoE session validation and creation. The IPoE session gets created, as long as the configuration exists and the subscriber information is validated successfully.

Configuring IPv6 IPoE Subscriber Interface

Perform this task to configure IPoE subscriber interface.

SUMMARY STEPS

1. **configure**
2. **pool vrf** *name* **ipv6** *pool_name*
3. **address-range** *first_ipv6_address last_ipv6_address*
4. **pool vrf** *name* **ipv6** *pool_name*
5. **prefix-length** *length*
6. **prefix-range** *first_ipv6_address last_ipv6_address*
7. **commit**
8. **dhcp ipv6**
9. **interface** *type interface-path-id* **server profile** *profile_name*
10. **profile** *server_profile_name* **server**
11. **prefix-pool** *pool_name*
12. **address-pool** *pool_name*
13. **commit**
14. **dhcp ipv6**
15. **interface** *type interface-path-id* **proxy profile** *profile_name*
16. **profile** *server_profile_name* **proxy**
17. **helper-address vrf** *vrf_name ipv6_address*

18. **commit**
19. **dynamic-template type ipsubscriber** *dynamic_template_name*
20. **ipv6 enable**
21. **dhcpv6 address-pool** *pool_name*
22. **dhcpv6 delegated-prefix-pool** *pool_name*
23. **commit**
24. **class-map type control subscriber match-all** *class-map_name*
25. **match protocol dhcpv6**
26. **end-class-map**
27. **policy-map type control subscriber** *class-map_name*
28. **event session-start match-first**
29. **class type control subscriber** *class_name* **do-all**
30. *sequence_number* **activate dynamic-template** *dynamic-template_name*
31. **end-policy-map**
32. **commit**
33. **interface** *type interface-path-id*
34. **ipv4 address** *ipv4_address*
35. **ipv6 address** *ipv6_address*
36. **ipv6 enable**
37. **service-policy type control subscriber** *name*
38. **ipsubscriber ipv6 l2-connected**
39. **initiator dhcp**
40. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**pool vrf** *name* **ipv6** *pool_name* **Example:** `RP/0/RSP0/CPU0:router(config)# pool vrf default ipv6 pool1`	Configures the distributed address pool service.
Step 3	**address-range** *first_ipv6_address last_ipv6_address* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# address-range 2201:abcd:1234:2400:f800::1 2201:abcd:1234:2400:f800::fff`	Configures the address-range.
Step 4	**pool vrf** *name* **ipv6** *pool_name* **Example:** `RP/0/RSP0/CPU0:router(config)# pool vrf default ipv6 pool2`	Configures the distributed address pool service.
Step 5	**prefix-length** *length* **Example:**	Specifies the prefix-length to be used.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-length 92`	
Step 6	**prefix-range** *first_ipv6_address last_ipv6_address* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-range 3301:1ab7:2345:1200:f800:: 3301:1ab7:2345:1200:f800:fff0::`	Specifies the prefix-range for allocation.
Step 7	**commit**	
Step 8	**dhcp ipv6** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	Configures DHCP for IPv6 and enters the DHCPv6 configuration mode.
Step 9	**interface** *type interface-path-id* **server profile** *profile_name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# interface Bundle-Ether1.1 server profile foo`	Associates a DHCPv6 proxy configuration profile to an IPv6 interface.
Step 10	**profile** *server_profile_name* **server** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# profile foo server`	Creates a DHCPv6 server profile and enters the DHCPv6 server profile sub-configuration mode.
Step 11	**prefix-pool** *pool_name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# prefix-pool pool2`	Configures a prefix pool in a server profile.
Step 12	**address-pool** *pool_name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# address-pool pool1`	Configures an address pool in the server profile.
Step 13	**commit**	
Step 14	**dhcp ipv6** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	Configures DHCP for IPv6 and enters the DHCPv6 configuration mode.
Step 15	**interface** *type interface-path-id* **proxy profile** *profile_name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# interface Bundle-Ether1.1 proxy profile foo`	Associates a DHCPv6 proxy configuration profile to an IPv6 interface.

	Command or Action	Purpose
Step 16	**profile** *server_profile_name* **proxy** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# profile foo proxy`	Creates a DHCPv6 server profile and enters the DHCPv6 server profile sub-configuration mode.
Step 17	**helper-address vrf** *vrf_name* *ipv6_address* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-proxy-profile)# helper-address vrf my-server-vrf 1:1:1::1`	Configures DHCPv6 address as a helper address to the proxy. **Note** The helper address can be configured only under the proxy profile and proxy profile class sub-modes.
Step 18	**commit**	
Step 19	**dynamic-template type ipsubscriber** *dynamic_template_name* **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template type ipsubscriber dhcpv6_temp`	Configures the dynamic template of type ipsubscriber and enters the dynamic template type configuration mode.
Step 20	**ipv6 enable** **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 enable`	Enables IPv6 on an interface.
Step 21	**dhcpv6 address-pool** *pool_name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# dhcpv6 address-pool pool3`	Configures DHCPv6 address pool.
Step 22	**dhcpv6 delegated-prefix-pool** *pool_name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# dhcpv6 delegated-prefix-pool pool4`	Configures DHCPv6 delegated prefix pool.
Step 23	**commit**	
Step 24	**class-map type control subscriber match-all** *class-map_name* **Example:** `RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-all dhcpv6_class`	Configures the class map control subscriber with a match-any criteria.
Step 25	**match protocol dhcpv6** **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# match protocol dhcpv6`	Configures match criteria for the class configured in the earlier step.

	Command or Action	Purpose
Step 26	**end-class-map** **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	Configures the end class map.
Step 27	**policy-map type control subscriber** *class-map_name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type control subscriber dhcpv6-policy`	Configures the subscriber control policy map.
Step 28	**event session-start match-first** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-first`	Configures the policy event with the match-first criteria.
Step 29	**class type control subscriber** *class_name* **do-all** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber dhcpv6_class do-all`	Configures the class map control subscriber with a match-any criteria.
Step 30	*sequence_number* **activate dynamic-template** *dynamic-template_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 20 activate dynamic-template dhcpv6_temp`	Activates actions related to dynamic template.
Step 31	**end-policy-map** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# end-policy-map`	Configures the end policy map.
Step 32	**commit**	
Step 33	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether1.1`	Configures an interface and enters the interface configuration mode.
Step 34	**ipv4 address** *ipv4_address* **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 11.11.11.2 255.255.255.0`	Configures the ipv4 address on an interface.
Step 35	**ipv6 address** *ipv6_address* **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv6 address 11:11:11::2/64`	Configures the ipv6 address on an interface.
Step 36	**ipv6 enable**	Enables IPv6 on an interface.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router(config-if)# ipv6 enable`	
Step 37	**service-policy type control subscriber** *name* **Example:** `RP/0/RSP0/CPU0:router(config-if)# service-policy type control subscriber dhcpv6_policy`	Associates a subscriber control service policy to the interface.
Step 38	**ipsubscriber ipv6 l2-connected** **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipsubscriber ipv6 l2-connected`	Enables l2-connected IPv6 subscriber.
Step 39	**initiator dhcp** **Example:** `RP/0/RSP0/CPU0:router(config-if-ipsub-ipv6-l2conn)# initiator dhcp`	Configures IPv6 subscriber initiator.
Step 40	**commit**	

Configuring IPv6 IPoE Subscriber Interface: An example

```
configure
pool vrf default ipv6 pool1
 address-range 2201:abcd:1234:2400:f800::1 2201:abcd:1234:2400:f800::fff

pool vrf default ipv6 pool2
prefix-length 92
prefix-range 3301:1ab7:2345:1200:f800:: 3301:1ab7:2345:1200:f800:fff0::

dhcp ipv6
 interface GigabitEthernet0/3/0/0 server profile foo
 profile foo server
  prefix-pool pool2
  address-pool pool1
!
!
end

configure
dhcp ipv6
 interface GigabitEthernet0/3/0/0 proxy profile foo
 profile foo proxy
  helper address <v6 address of the server
!
!
dynamic-template type ipsubscriber dhcpv6_temp
  ipv6 enable
  dhcpv6 address-pool pool3
  dhcpv6 delegated-prefix-pool pool4
 !
!
!
class-map type control subscriber match-all dhcpv6_class
```

```
 match protocol dhcpv6
 end-class-map
!
policy-map type control subscriber dhcpv6_policy
 event session-start match-first
  class type control subscriber dhcpv6_class do-all
   20 activate dynamic-template dhcpv6_temp
!
!
end

configure
interface GigabitEthernet0/3/0/0
 ipv4 address 11.11.11.2 255.255.255.0
 ipv6 address 11:11:11::2/64
 ipv6 enable
 service-policy type control subscriber dhcpv6_policy
 ipsubscriber ipv6 l2-connected
  initiator dhcp
!
!
end
```

end

IPv6 PPPoE Subscriber Support

The PPPoE subscriber interfaces establish a PPP link with the subscriber, which is used for authentication and address assignment. The DHCPv6 server assigns the address or prefix to the PPPoE subscriber. Because the PPPoE subscriber interfaces are created dynamically, the DHCPv6 profile is applied to all the PPPoE interfaces created on the router, and not just a single PPPoE interface.

To enable PPPoE subscriber support, you have to configure the DHCPv6 profile globally or on all PPPoE interfaces. For more information, see Configuring IPv6 PPPoE Subscriber Interfaces, on page 148.

Configuring IPv6 PPPoE Subscriber Interfaces

Perform this task to configure PPPoE subscriber interfaces.

SUMMARY STEPS

1. **configure**
2. **dynamic-template type ppp** *dynamic_template_name*
3. **ppp authentication chap**
4. **ppp ipcp peer-address pool** *pool_name*
5. **ipv4 unnumbered** *interface-type interface-path-id*
6. **ipv6 enable**
7. **commit**
8. **class-map type control subscriber match-any** *class-map_name*
9. **match protocol ppp**
10. **end-class-map**
11. **commit**
12. **class-map type control subscriber match-all** *class-map_name*
13. **match protocol dhcpv6**
14. **end-class-map**

15. **commit**
16. **policy-map type control subscriber** *policy_name*
17. **event session-start match-first**
18. **class type control subscriber** *name* **do-all**
19. *sequence_number* **activate dynamic-template** *dynamic-template_name*
20. **end-policy-map**
21. **policy-map type control subscriber** *policy_name*
22. **event session-start match-all**
23. **class type control subscriber** *name* **do-all**
24. *sequence_number* **activate dynamic-template** *dynamic-template_name*
25. **end-policy-map**
26. **commit**
27. **interface** *type interface-path-id*
28. **description** *LINE*
29. **ipv6 enable**
30. **service-policy type control subscriber** *name*
31. **encapsulation dot1q 801**
32. **ipsubscriber ipv6 l2-connected**
33. **initiator dhcp**
34. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dynamic-template type ppp** *dynamic_template_name* **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template type ppp ppp_pta_template`	Configures the dynamic template of type ppp and enters the dynamic template type configuration mode.
Step 3	**ppp authentication chap** **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ppp authentication chap`	Configures challenge handshake authentication protocol (chap) and sets PPP link authentication method.
Step 4	**ppp ipcp peer-address pool** *pool_name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ppp ipcp peer-address pool p1`	Sets ipcp negotiation options and sets the peer address configuration option for the peer-address pool.
Step 5	**ipv4 unnumbered** *interface-type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv4 unnumbered Loopback 1`	Enables IPv4 processing without an explicit address for an interface.

	Command or Action	Purpose
Step 6	**ipv6 enable** **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 enable`	Enables IPv6 on an interface.
Step 7	**commit**	
Step 8	**class-map type control subscriber match-any** *class-map_name* **Example:** `RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-any pta_class`	Configures the class map control subscriber with a match-any criteria.
Step 9	**match protocol ppp** **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# match protocol ppp`	Configures match criteria for the class configured in the earlier step.
Step 10	**end-class-map** **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	Configures the end class map.
Step 11	**commit**	
Step 12	**class-map type control subscriber match-all** *class-map_name* **Example:** `RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-all ipoe_test`	Configures the class map control subscriber with a match-all criteria.
Step 13	**match protocol dhcpv6** **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# match protocol dhcpv6`	Configures match criteria for the class configured in the earlier step.
Step 14	**end-class-map** **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	Configures the end class map.
Step 15	**commit**	
Step 16	**policy-map type control subscriber** *policy_name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type control subscriber policy1`	Configures the subscriber control policy map.
Step 17	**event session-start match-first** **Example:**	Configures the policy event with the match-first criteria.

	Command or Action	Purpose
	RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-first	
Step 18	**class type control subscriber** *name* **do-all** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class type control subscriber ipoe_test1 do-all`	Configures the policy event with the match-first criteria.
Step 19	*sequence_number* **activate dynamic-template** *dynamic-template_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 24 activate dynamic-template v6_test1`	Activates actions related to dynamic template.
Step 20	**end-policy-map** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# end-policy-map`	Configures the end policy map.
Step 21	**policy-map type control subscriber** *policy_name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type control subscriber policy1`	Configures the subscriber control policy map.
Step 22	**event session-start match-all** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-all`	Configures the policy event with the match-all criteria.
Step 23	**class type control subscriber** *name* **do-all** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class type control subscriber pta_class do-all`	Configures the policy event with the match-first criteria.
Step 24	*sequence_number* **activate dynamic-template** *dynamic-template_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 1 activate dynamic-template ppp_pta_template`	Activates actions related to dynamic template.
Step 25	**end-policy-map** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# end-policy-map`	Configures the end policy map.
Step 26	**commit**	

	Command or Action	Purpose
Step 27	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface BundleEther1.1`	Configures an interface and enters the interface configuration mode.
Step 28	**description** *LINE* **Example:** `RP/0/RSP0/CPU0:router(config-if)# description IPoE`	Sets the description for the above configured interface.
Step 29	**ipv6 enable** **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv6 enable`	Enables IPv6 on an interface.
Step 30	**service-policy type control subscriber** *name* **Example:** `RP/0/RSP0/CPU0:router(config-if)# service-policy type control subscriber ipoe1`	Associates a subscriber control service policy to the interface.
Step 31	**encapsulation dot1q 801** **Example:** `RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 801`	Enables encapsulated 802.1Q VLAN configuration.
Step 32	**ipsubscriber ipv6 l2-connected** **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipsubscriber ipv6 l2-connected`	Enables l2-connected IPv6 subscriber.
Step 33	**initiator dhcp** **Example:** `RP/0/RSP0/CPU0:router(config-if-ipsub-ipv6-l2conn)# initiator dhcp`	Configures IPv6 subscriber initiator.
Step 34	**commit**	

Configuring IPv6 PPPoE Subscriber Interfaces: An example

```
configure
dynamic-template
type ppp PPP_PTA_TEMPLATE
ppp authentication chap
ppp ipcp peer-address pool ADDRESS_POOL
ipv4 unnumbered Loopback0
ipv6 enable
!
type ipsubscriber v6_test1
ipv6 enable
!
```

```
!
class-map type control subscriber match-any PTA_CLASS
match protocol ppp
end-class-map
!
class-map type control subscriber match-all ipoe_test1
match protocol dhcpv6
end-class-map
!
policy-map type control subscriber ipoe1
event session-start match-first
class type control subscriber ipoe_test1 do-all
24 activate dynamic-template v6_test1
!
!
end-policy-map
!
policy-map type control subscriber POLICY1
event session-start match-all
class type control subscriber PTA_CLASS do-all
1 activate dynamic-template PPP_PTA_TEMPLATE
!
!
end-policy-map
!
interface Bundle-Ether2.801
description IPoE
ipv6 enable
service-policy type control subscriber ipoe1
encapsulation dot1q 801
ipsubscriber ipv6 l2-connected
initiator dhcp
```

Ambiguous VLAN Support

An Ambiguous VLAN is configured with a range or group of VLAN IDs. The subscriber sessions created over ambiguous VLANs are identical to subscribers over regular VLANs that support all regular configurations such as policy-map, VRFs, QoS, and ACL. Multiple subscribers can be created on a particular VLAN ID as long as they contain a unique MAC address. Ambiguous VLANs enhance scalability by reducing the need for configuring multiple access interfaces.

To enable DHCPv6 support, ambiguous VLANs are unnumbered on top of the bundle interface.

Note The ambiguous VLANs are named exactly the same way as regular VLANs. The ambiguous VLANs are considered Layer 3 interfaces in contrast to EFP ranges allowed for l2transport interface.

When DHCPv6 Server receives a SOLICIT message on the ambiguous VLAN interface, the VLAN IDs are extracted from the received packet and used for authenticating the subscriber with the client related information.

When an interface configuration is changed from ambiguous to non-ambiguous or vice-versa or Ambiguous VLAN range is changed, then all existing client bindings for the Ambiguous VLAN are cleared.

For more information on configuring ambiguous VLAN, see Configuring Ambiguous VLANs, on page 153.

Configuring Ambiguous VLANs

Perform this task to configure ambiguous vlans.

Note There is no DHCP-specific configuration required for ambiguous VLANs.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. Use any of these encapsulations to configure encapsulated ambiguous VLANs:
    - **encapsulation ambiguous** { **dot1q** | **dot1ad** } {**any** | *vlan-range* }
    - **encapsulation ambiguous dot1q** *vlan-id* **second-dot1q** { **any** | *vlan-range* }
    - **encapsulation ambiguous dot1q any second-dot1q** { **any** | *vlan-id* }
    - **encapsulation ambiguous dot1ad** *vlan-id* **dot1q** { **any** | *vlan-range* }
    - **encapsulation ambiguous dot1q** *vlan-range* **second-dot1q any**
    - **encapsulation ambiguous dot1ad** *vlan-range* **dot1q any**
4. **ipv4 | ipv6address** *source-ip-address destination-ip-address*
5. **service-policy type control subscriber** *policy_name*
6. **ipsubscriber** { **ipv4** | **ipv6** } **l2-connected**
7. **initiator dhcp**
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether100.12`	Configures the interface and enters the interface configuration mode.
Step 3	Use any of these encapsulations to configure encapsulated ambiguous VLANs: • **encapsulation ambiguous** { **dot1q** | **dot1ad** } {**any** | *vlan-range* } • **encapsulation ambiguous dot1q** *vlan-id* **second-dot1q** { **any** | *vlan-range* } • **encapsulation ambiguous dot1q any second-dot1q** { **any** | *vlan-id* } • **encapsulation ambiguous dot1ad** *vlan-id* **dot1q** { **any** | *vlan-range* } • **encapsulation ambiguous dot1q** *vlan-range* **second-dot1q any** • **encapsulation ambiguous dot1ad** *vlan-range* **dot1q any** **Example:**	Configures IEEE 802.1Q VLAN configuration. The *vlan-range* can be given in comma-separated, or hyphen-separated format, or a combination of both, as shown in the examples. **Note** Although **encapsulation ambiguous dot1ad** is supported, it is not commonly used in BNG deployments.

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q any`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q 14 second-dot1q 100-200`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q any second-dot1q any`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1ad 14 dot1q 100,200,300-400`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q 1-1000 second-dot1q any`	

Step 4	**ipv4 | ipv6address** *source-ip-address destination-ip-address*
	Example:
	`RP/0/RSP0/CPU0:router(config-if)# ipv4 address 2.1.12.1 255.255.255.0`
	`RP/0/RSP0/CPU0:router(config-if)# ipv6 address 1:2:3::4 128`
		Configures the IPv4 or IPv6 protocol address.

Step 5	**service-policy type control subscriber** *policy_name* **Example:** `RP/0/RSP0/CPU0:router(config-if)# service-policy type control subscriber PL1`	Applies a policy-map to an access interface where the policy-map was previously defined with the specified PL1 policy_name.

Step 6	**ipsubscriber** { **ipv4** | **ipv6** } **l2-connected**
	Example:
	`RP/0/RSP0/CPU0:router(config-if)# ipsubscriber ipv4 l2-connected`
	`RP/0/RSP0/CPU0:router(config-if)# ipsubscriber ipv6 l2-connected`
		Enables l2-connected IPv4 or IPv6 IP subscriber.

Step 7	**initiator dhcp** **Example:** `RP/0/RSP0/CPU0:router(config-if)# initiator dhcp`	Enables initiator DHCP on the IP subscriber.
Step 8	**commit**	

Configuring Ambiguous VLANs: An example

```
configure
interface Bundle-Ether100.12
encapsulation ambiguous dot1q 14 second-dot1q any
ipv4 address 2.1.12.1 255.255.255.0
service-policy type control subscriber PL1
ipsubscriber ipv4 l2-connected
initiator dhcp
!
!
```

```
end
```

DHCPv6 Address or Prefix Pool

An address or prefix pool represents a pool of available address or prefixes from which a delegating router assigns an address or delegates a prefix to the requesting router. The Distributed Address Pool Service (DAPS) manages and maintains address or prefix pools for DHCPv6.

DHCPv6 Prefix Delegation involves a delegating router selecting a prefix and delegating it on a temporary basis to a requesting router. The delegating router assigns the address or delegates the prefix from the address pool or prefix pool to the requesting router.

For more information about configuring DHCPv6 address or prefix pool, see Configuring IPv6 Address or Prefix Pool Name, on page 156.

Configuring IPv6 Address or Prefix Pool Name

Perform this task to configure IPv6 address or prefix pool name under dynamic template configuration mode.

SUMMARY STEPS

1. **configure**
2. **dynamic-template**
3. **type ipsubscriber** *dynamic-template_name*
4. **dhcpv6 delegated-prefix-pool** *pool-name*
5. **commit**
6. **type ppp** *dynamic-template_name*
7. **dhcpv6 address-pool** *pool-name*
8. **commit**
9. **type ipsubscriber** *dynamic-template_name*
10. **dhcpv6 address-pool** *pool-name*
11. **commit**
12. **ipv6 nd framed-prefix-pool** *pool-name*
13. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dynamic-template** **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template`	Enables dynamic template configuration.
Step 3	**type ipsubscriber** *dynamic-template_name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ipsubscriber ipv6-sub-template`	Configures dynamic template of type ipsubscriber and enters the dynamic-template type configuration mode.

	Command or Action	Purpose
Step 4	**dhcpv6 delegated-prefix-pool** *pool-name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# dhcpv6 delegated-prefix-pool mypool`	Configures IPv6 subscriber dynamic template with prefix-delegation pool.
Step 5	**commit**	
Step 6	**type ppp** *dynamic-template_name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ppp ipv6-sub-template`	Configures dynamic template of type ppp.
Step 7	**dhcpv6 address-pool** *pool-name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# dhcpv6 address-pool my-pppoe-addr-pool`	Configures IPv6 address pool for PPPoE subscribers.
Step 8	**commit**	
Step 9	**type ipsubscriber** *dynamic-template_name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ipsubscriber my-ipv6-template`	Configures dynamic template of type ipsubscriber and enters the dynamic-template type configuration mode.
Step 10	**dhcpv6 address-pool** *pool-name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# dhcpv6 address-pool my-ipsub-addr-pool`	Configures IPv6 address pool for IPoE subscribers.
Step 11	**commit**	
Step 12	**ipv6 nd framed-prefix-pool** *pool-name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# framed-prefix-pool my-slaac-pool`	Configures prefix pool to be used by SLAAC only.
Step 13	**commit**	

Configuring IPv6 Address or Prefix Pool Name: An example

```
configure
dynamic-template
type ipsubscriber ipv6-sub-template
dhcpv6 delegated-prefix-pool mypool
end
dynamic-template
type ppp ipv6-sub-template
dhcpv6 address-pool my-pppoe-addr-pool
!
type ipsubscriber my-ipv6-template
```

```
dhcpv6 address-pool my-ipsub-addr-pool
!!
ipv6 nd framed-prefix-pool my-slaac-pool
end
!!
```

DHCPv6 Dual-Stack Lite Support

Dual-Stack Lite (DS-Lite) is a technique for providing complete support for both IPv4 and IPv6 internet protocols, both in hosts and router. Dual-Stack Lite enables a broadband service provider to share IPv4 addresses among customers by combining two technologies: IP in IP (IPv4- in-IPv6) and Network Address Translation (NAT).

The DS-Lite feature contains two components: Basic Bridging Broad Band (B4) and Address Family Transition Router (AFTR).

The B4 element is a function implemented on a dual-stack-capable node, either a directly connected device or a CPE that creates a tunnel to an Address Family Transition Router (AFTR). On the other hand, an AFTR element is the combination of an IPv4-in-IPv6 tunnel endpoint and an IPv4-IPv4 NAT implemented on the same node. A DS-Lite B4 element uses a DHCPv6 option to discover the IPv6 address of its corresponding AFTR location.

For more information about configuring AFTR for DS-Lite, see Configuring AFTR Fully Qualified Domain Name for DS-Lite, on page 158.

Configuring AFTR Fully Qualified Domain Name for DS-Lite

Perform this task to configure AFTR fully qualified domain name for DS-Lite.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv6**
3. **profile** *server_profile_name* **server**
4. **aftr-name** *aftr_name*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv6** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	Configures DHCP for IPv6 and enters the DHCPv6 configuration mode.
Step 3	**profile** *server_profile_name* **server** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# profile my-server-profile server`	Configures DHCPv6 server profile and enters the DHCPv6 server profile sub-configuration mode.

	Command or Action	Purpose
Step 4	**aftr-name** *aftr_name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# aftr-name aftr-server.example.com`	Configures the AFTR Fully Qualified Domain Name option, in the server profile mode, for the DS-Lite support.
Step 5	**commit**	

Configuring AFTR Fully Qualified Domain Name for DS-Lite: An example

```
configure
dhcp ipv6
profile my-server-profile server
aftr-name aftr-server.example.com
end
!!
```

VRF Awareness in DHCPv6

VRF Awareness is the ability of DHCPv6 Server or Proxy to support multiple clients in different VPNs where the same IP address is assigned to clients on differing VPNs. The IPv6 addresses in a VRF is independent from IPv6 addresses in an another VRF. It is not mandatory to have same prefix/address in multiple VRFs.

For more information about defining VRF in a dynamic template, see Defining VRF in a Dynamic Template, on page 159.

Defining VRF in a Dynamic Template

Perform this task for defining VRF in a dynamic template. The IPv6 addresses in a VRF is independent from IPv6 addresses in an another VRF. It is not mandatory to have same prefix or address in multiple VRFs.

SUMMARY STEPS

1. **configure**
2. **dynamic-template**
3. **type ipsubscriber** *dynamic-template_name*
4. **vrf** *vrf_name*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dynamic-template** **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template`	Enables dynamic template configuration.

	Command or Action	Purpose
Step 3	**type ipsubscriber** *dynamic-template_name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ipsubscriber ipv6-sub-template`	Configures dynamic template of type ipsubscriber and enters the dynamic template type configuration mode.
Step 4	**vrf** *vrf_name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# vrf vrf1`	Sets the VRF in which the interface operates.
Step 5	**commit**	

Defining VRF in a Dynamic Template: An example

```
configure
dynamic-template
type ipsubscriber ipv6-sub-template
vrf vrf1
end
!!
```

Rapid commit

The **rapid-commit** command aids to enable or disable the rapid commit option of the DHCP server. Enabling it renders the DHCPv6 server to use the two message exchange feature to address/prefix an assignment. Including the rapid commit option in the `SOLICIT` message and enabling the same in the server profile, enables the server to respond with the `REPLY` message. Else, it follows the normal four message exchange procedure to assign address/prefix an assignment.

Note By default, the rapid commit option is disabled.

Example:

```
RP/0/RSP0/CPU0:router(config)# dhcp ipv6
RP/0/RSP0/CPU0:router(config-dhcpv6)# profile my-server-profile server
RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# rapid-commit
```

Packet Handling on Subscriber Interfaces

This section describes how subscriber interfaces are supported in certain special cases. These special cases include L3 forwarded interfaces. As a result, this support is applicable only to PPP over Ethernet PPP Termination and Aggregation (PPPoE PTA) and IPoE sessions.

Most subscriber data packets are forwarded directly by the network processing unit (NPU). There are certain special cases where the NPU does not completely handle the data packet. These special cases are handled by the CPU, and go through an internal interface created for this purpose. This internal interface is named the Subscriber Interface or SINT. SINT is an aggregate interface, which is used by all packets punted on subscriber interfaces. There is one SINT for each node. When the BNG package is installed, by default the SINT is created. The SINT interfaces are needed for punt-inject of packets on subscriber interfaces.

These special cases are supported for both IPoE and PPPoE PTA:

Note These special cases do not apply to PPPoE L2TP, because it is an L2 service.

- Ping to and from subscriber

  BNG allows the receiving of a ping request from both IPoE and PPPoE PTA subscriber interfaces; this is consistent with other non-BNG interface types as well. Similarly, BNG also allows the sending of a ping request to both IPoE and PPPoE PTA subscriber interfaces. This includes:

  - various lengths of ping packets including lengths exceeding the subscribers MTU size
  - subscriber in the default and private VRFs
  - various ping options such as type of service, DF set, and verbose

  BNG also supports receiving a ping request from both IPv4 and IPv6 subscribers.

  Note Excessive Punt Flow Trap feature should be disabled when sending a high rate of pings to, or from subscriber interfaces.

- Option Handling

  BNG supports handling IP options; this is consistent with non-BNG interface types. These are punted from the NPU to the CPU. These go through the SINT interface and are handled by the appropriate application.

- Support for traceroute, PMTU discovery, ICMP unreachable

  - BNG supports sending ICMP for packets that are received from or destined to a PPPoE or IP subscriber interface that cannot be forwarded. This functionality is similar to other non-BNG subscriber interfaces.
  - BNG supports PMTU, in which BNG sends ICMPs, when a packet is destined to a subscriber interface, but the packet exceeds the subscriber MTU and the DF bit is set.
  - BNG supports sending ICMPs when packets to (egress ACL) or from (ingress ACL) the subscriber interface are denied due to the ACL. If the ACL is configured do both deny and log, then the packets get dropped, but no ICMP is generated.
  - BNG supports traceroute functionality that enables sending an ICMP when the time to live (TTL) of the packet is exceeded.
  - BNG supports traceroute functionality for both IPv4 and IPv6 subscribers.

- Fragmentation

  BNG does not support fragmentation of packets destined to the PPPoE or IP subscriber interfaces.

> **Caution** In Cisco IOS XR, fragmentation is handled by linecard (LC) CPU or route processor (RP) CPU. All packets requiring fragmentation are policed by local packet transport service (LPTS), to a maximum of 2500 packets per second (pps) for each network processing unit (NPU).
>
> The fragmentation path is supported only in software, and fragmented packets skip all features, including subscriber features, QoS, ACL and so on. Therefore, irrespective of BNG, it should not be used as a general forwarding path.
>
> BNG over Pseudowire Headend does not support fragmentation.

Restrictions

These restrictions apply to implementing subscriber interfaces:

- During an ACL logging, packets are punted to CPU, and BNG interfaces are directed to the SINT interface. The SINT interface drops these log packets because the system does not support ACL Logging on BNG interfaces.
- IPv6 Ping and traceroute functions should use both the CPE and BNG routers global addresses. IPv6 Ping and traceroute functions using link local address does not work in all cases.
- Logging on subscriber ACLs is not supported.

IPv6 Neighbor Discovery

The IPv6 neighbor discovery process uses Internet Control Message Protocol (ICMP) messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and track neighboring routers.

The IPv6 static cache entry for neighbor discovery feature allows static entries to be made in the IPv6 neighbor cache. Static routing requires an administrator to manually enter IPv6 addresses, subnet masks, gateways, and corresponding Media Access Control (MAC) addresses for each interface of each device into a table. Static routing enables more control but requires more work to maintain the table. The table must be updated each time routes are added or changed.

The different message types in neighbor discovery are:

- IPv6 Neighbor Solicitation Message: A value of 135 in the Type field of the ICMP packet header identifies a neighbor solicitation message. Neighbor solicitation messages are sent on the local link when a node wants to determine the link-layer address of another node on the same local link.
- IPv6 Router Advertisement Message: Router advertisement (RA) messages, which have a value of 134 in the Type field of the ICMP packet header, are periodically sent out of each configured interface of an IPv6 device.

  Ambiguous VLAN does not have association with any particular VLAN, and therefore, a unicast router advertisement message has to be sent out for ambiguous VLAN interfaces. To enable IPv6 unicast router advertisement, you must use the **ipv6 nd ra-unicast** command in the dynamic template configuration mode.

Note From Cisco IOS XR Release 5.1.0 and later, it is mandatory to configure **ipv6 enable** command under the bundle access-interface, in order to send RA messages out of BNG.

- IPv6 Neighbor Redirect Message: A value of 137 in the type field of the ICMP packet header identifies an IPv6 neighbor redirect message. Devices send neighbor redirect messages to inform hosts of better first-hop nodes on the path to a destination.

In BNG, IPv6 neighbor discovery supports both IPoE and PPPoE sessions. IPv6 neighbor discovery provides Stateless Address Auto Configuration (SLAAC), which is used for assigning a prefix to the PPPoE subscriber.

Line Card Subscribers

BNG supports line card (LC) subscribers which are based on physical access interfaces. This support is in addition to supporting route processor (RP) subscribers, which are based on bundle access-interfaces. Apart from route switch processor (RSP), line cards also support session termination and control plane protocols. For LC subscribers, both control and data planes run on the same node and share the same CPU resource. In contrast, for bundle subscribers, the control plane runs completely on RSP, and the data plane runs completely on LC.

The number of LC subscribers sessions scales linearly with the number of line cards in the system. The maximum number of sessions for each LC is 64000. As more line cards are added to the system, the maximum number of sessions in the system reaches a multiple of 64000 subscribers, the multiplier being the number of line cards.

The calls-per-second (CPS) achieved for each chassis scales almost linearly with the number of line cards in the system. Linearity is not achieved for CPS because of the congestion in the communication channel, arising out of the large number of notifications sent out from LC to RSP.

External Interaction for LC Subscribers

As part of LC subscriber support, there are various interactions directly between LC and external servers such as RADIUS and DHCP servers. These interactions change the way how load balancing is done and the way CoA is handled.

Load Balancing

Because each LC control plane functions independently, with LC subscribers, any global configuration of RADIUS and DHCP servers does not result in load-balanced usage. It is possible that all LCs end up using the same RADIUS server. As a result, the user needs to carry out manual load balancing. This is done by creating different AAA groups and method lists using different sets of RADIUS servers, then assigning the AAA groups to different service profiles, and finally assigning these different service profiles to the access interfaces on different LCs. Similarly, for DHCP servers, the access interfaces on different LCs should have different profiles, each pointing to different DHCP servers.

Interaction with RADIUS Server

With the distributed model of interacting with RADIUS, the RADIUS client on BNG can be configured in two different ways. Either the entire BNG router shows up as one BNG to the RADIUS server (**NAS-IP-Address**), or each LC appears as a different router. Currently, the CoAs can be handled only by the iEdge on the RSP. Each LC appearing as its NAS is not supported.

Address Pools

It is preferable to provide different address pools to different LCs so that they work completely independent of each other, without the need to perform significant messaging across nodes.

Benefits and Restrictions of Line Card Subscribers

Benefits of line card subscribers

These are some of the benefits of LC subscribers:

- Subscribers built on bundle interfaces and line card physical interfaces can co-exist on the same router.
- Significant gain in performance because the control plane is distributed to multiple LCs. In aggregate, the entire chassis reaches much higher scale and performance than RSP-based subscribers.
- Higher fault isolation on the router. The control plane runs in a distributed manner and therefore, failure of certain LCs does not affect subscriber sessions on other LCs in the system. In such cases, only the subscriber sessions built on that particular LC is lost.
- Although the CPS achieved on a single LC is lower than the CPS achieved for RSP or Bundle subscribers, LC subscribers overcome the memory usage limit and CPS limit of RSP-based subscribers.
- Provide enhanced multi-service edge (MSE) capability for the ASR9K router, by freeing up the CPU and memory resources on the centralized route processor (RP).

Restrictions of line card subscribers

LC subscriber support in BNG is subjected to these restrictions:

- Bundles are not supported with LC subscribers.
- LC subscribers support features that are available on bundle subscribers, except for a few features such as Parameterized QoS, multicast, and service accounting. If these features are required for specific subscribers, then those subscribers must be built on bundle interfaces.
- Routed subscriber sessions are not supported on LC subscribers.
- Local DHCPv4 Server feature is not supported over LC.

Note From Cisco IOS XR Software Release 5.3.2 and later, features such as Parameterized QoS and service accounting are supported for LC subscribers as well.

High Availability for Line Card Subscribers

The high availability (HA) for line card subscribers is different from that for subscribers built on bundle interfaces because the subscribers are built on LCs. This table details the HA features of LC subscribers and bundle subscribers:

Table 7: High Availability for LC Subscribers and Bundle Subscribers

HA Feature	Plane	Bundle Subscribers	Line Card Subscribers
Process restart	control	Subscriber session state is maintained. New subscriber bring up is delayed by a short time, depending on the component being restarted.	Behavior is the same as for bundle subscribers.
	data	No impact to traffic.	No impact to traffic.
LC online insertion and removal (OIR)	control	No impact with multi-member bundles. Because control packet is not received, control plane cannot function with single member bundles. Session state is not lost because it is stored in RSP.	Control plane is down for new sessions, and all session states are lost for existing sessions. After LC OIR, the LC sessions are restored using DHCP shadow bindings in RP.
	data	No impact with multi-member bundles. Data traffic is lost with single member bundles. Session state is not lost.	All traffic is lost
RP failover	control	Significant quiet time (currently more than 10 minutes) is expected before new sessions can be setup. Existing session state is not lost.	Very small impact (approximately 10 seconds) before new sessions can be setup; the delay is in connecting to RSP based servers, like RIB. Existing session state is not lost.
	data	No impact to traffic.	No impact to traffic.

Static Sessions

BNG supports interface-based static sessions, where all traffic belonging to a particular VLAN sub-interface is treated as a single session. These sessions are created or deleted, based on the configuration of static session on the sub-interface (access-interface). The session establishment is triggered by creating a static subscriber configuration on a sub-interface; the session termination is triggered by removing that configuration.

The number of static sessions that can be created in a router is the same as the number of Bundle VLAN interfaces that can be present in the router.

Static sessions are present only in the control plane, mainly to provide access to AAA, CoA, and dynamic templates. These sessions have the same flexibility as other kinds of sessions (such as DHCP-triggered sessions and packet-triggered sessions) from the perspective of AAA, CoA, and other dynamic configuration changes.

All forwarding and routing features for static sessions are programmed directly on the access-interface. Features such as Access Control List (ACL), Hierarchical Quality of Service (H-QoS), and Session Accounting are allowed to be configured through RADIUS or through dynamic template.

The IP address (and VRF, if used) for a static session is recommended to be configured on the access-interface itself (See the note below for the behavior of feature modification using BNG static sessions). All subnet interface addresses can be assigned to the subscribers in the case of switched Customer Premises Equipment (CPE). The Unicast Reverse Path Forwarding (uRPF) is also configured on the access-interface itself. Because the access-interface is like any other Layer 3 interface, it allows PE-CE routing protocols such as OSPF and BGP.

Note If any feature configured on the access-interface is modified using BNG, the existing configurations get removed from the access-interface, and they do not get restored automatically on removing the static session. For example, if an ACL is already present on the access-interface, and if another ACL is applied by BNG using the static session, the ACL on the access-interface does not get restored when the static session is removed. You must reconfigure the access-interface and add the ACL again, in such scenarios.

Another example of feature modification by BNG is, if a VRF (say, *vrf-blue*) is present on the access-interface, and if another VRF (say, *vrf-green*) is applied on the access-interface by BNG using the static session, *vrf-blue* on the access-interface is not restored when the static session is removed. The interface is set to the default VRF. You must reconfigure the access-interface and add the *vrf-blue* again, in such scenarios.

A static session is similar to a subscriber session, except for these differences:

- The CoA should explicitly have an account session ID because static session does not have MAC address or IP address identity attribute associated with it.
- The statistics of static session is the same as that of the access-interface on which it is configured.

Restrictions for static sessions

The interface-based static session in BNG is subject to these restrictions:

- Because all features are applied on the access-interface itself, all restrictions for feature programming on access-interface applies to static session too.
- The HTTP redirect feature is not supported for static session.
- Service accounting is not supported for static sessions.

Subscriber Session Limit

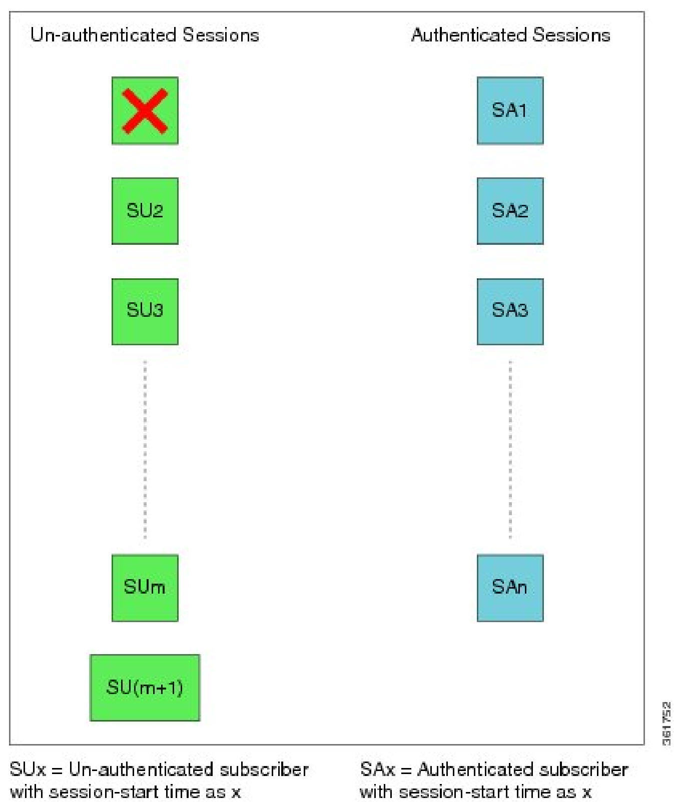
The subscriber session limit feature limits the total number of subscriber sessions in a BNG router. If a new subscriber session comes up after the router reaches the overall session limit, then the earliest un-authenticated session is deleted. If the router reaches the overall subscriber session limit and if all the sessions present in the router are authenticated sessions, then the request for a new session is rejected.

Typically sessions belonging to subscribers who do not have the intent of accessing the network services are typically un-authenticated sessions. Per-subscriber features do not apply to such sessions. Instead, they have the same set of features applied to all users. Generally, if the un-authenticated subscriber sessions do not authenticate themselves within a specific time, they are deleted using the un-auth timer mechanism.

The **subscriber session limit** command is used to apply the overall subscriber session limit in the BNG router.

This figure shows the scenario where a long-lived un-authenticated session is deleted, when a new un-authenticated session (m + 1) comes up after the router reaches the overall session limit. In this example, m+n is the overall session limit, where m is the number of un-authenticated sessions and n is the number of authenticated sessions. The behavior is the same for a new authenticated session (n + 1) too.

Figure 16: Subscriber Session Limit



BNG Subscriber Templates

BNG supports template interface based subscriber provisioning that defines an internal template interface for storing the feature information of each unique subscriber configuration, and reuses that information for feature programming of other subscribers having the same configuration. This reduces the inter-process communications (IPC), memory usage and CPU usage inside the system, thereby providing significant scale and performance improvements in BNG. The free memory available in the system can thus be utilized for enabling more services or for improving existing services on BNG, with full scale stability.

BNG subscriber templates feature is more beneficial in scale scenarios such as Service Provider Wi-Fi (SP Wi-Fi). Template interfaces are not recommended for scenarios where the configuration of template interface based feature is different for each subscriber, or in scenarios where only a few hundreds of subscribers use the similar configuration. Templates must be used or provisioned only if there are a few thousands of subscribers using similar configurations of template interface based features. There is no restriction on individual subscribers having different configurations for non-template interface based features.

Enabling BNG Subscriber Templates

Subscriber templates are enabled per access-interface in BNG. Use this command in interface configuration mode, to enable subscriber templates:

ipsubscriber subscriber-templates *max-templates*

Here, *max-templates* is the maximum number of templates on an access-interface.

This is an example of enabling subscriber templates on an access-interface in BNG:

```
interface Bundle-Ether1.10
 ipsubscriber subscriber-templates 5
!
```

You must clear all subscriber sessions on an access-interface before disabling the subscriber templates or before modifying the number of subscriber templates on that access-interface.

Feature Support for Subscriber Templates

These features are supported with subscriber templates:

- IPv4, IPv4-ACL, IPv6, IPv6-ACL.
- DHCP and packet-triggered sessions.
- RP subscribers.
- LC subscribers.
- High Availability - process restart and route processor fail over (RPFO).
- Scale scenarios.

These features are not supported with subscriber templates:

- PPPoE sessions.
- QoS and PBR.

Restrictions for BNG Subscriber Templates

The support for subscriber templates in BNG is subjected to these restrictions:

- Modifying the number of templates or removing the template configuration is not supported with subscribers provisioned on the access- interface.
- Modifying the encapsulation is not supported on access-interface having subscriber templates configured.
- Each line card (LC) has a micro-interface database (UIDB) limitation of 16 bits (that is, 65535 entries). For an expected scale of 64K subscriber interfaces on an LC, 1535 interfaces are remaining for the access-interfaces and template-interfaces. Provisioning of template-interfaces must be planned within these limits.

Verification of BNG Subscriber Templates

This table lists the verification commands for BNG subscriber templates configuration:

Command	Description
show ipsubscriber interface internal	Displays the internal information such as, *Template ID* (the template interface-handle referred by the subscriber session), of the IP subscriber interfaces.
show ipsubscriber template-interface [**access-interface** *interface-type interface-instance*] [**internal**]	Displays IP subscriber template interface information (brief, detailed or filtered based on the access-interface) such as template subscriber name, template subscriber ifhandle and so on.
show subscriber database session subscriber-label *subscriber-label*	Displays the subscriber database session information that includes the *Template Interface Id* field (this field indicates the subscriber template that is used by the session with the specified subscriber-label).
show subscriber database template [**parent-if-handle** *if-handle* | **parent-if-name** *interface-type interface-instance*]	Displays subscriber database information such as template ifhandle, session count and so on.
show subscriber running-config subscriber-label *label*	Displays the subscriber running configuration in BNG.

Along with these commands, the existing subscriber show commands can also be used to verify the configurations.

eBGP over PPPoE

The eBGP over PPPoE feature provides eiBGP multi-path support over BNG subscriber interfaces. This feature also provides load-balancing and allows service providers to offer L3VPN service with dynamic service provisioning. The label allocation mode used for this feature is **per-prefix**. The feature is supported for IPv4 and IPv6.

Benefits of eBGP over PPPoE

The eBGP over PPPoE feature provides eiBGP multi-path support with **per-prefix** label allocation mode. Currently, Cisco IOS XR supports three label allocation modes - per prefix, per-CE and per-VRF. The per-VRF mode does not provide multi-path support, and it may also cause forwarding loops during local traffic diversion. The per-CE mode does not support eBGP load balancing and BGP PIC functionality. Therefore, the per-prefix mode is chosen for this feature.

For sample topology and sample configurations for eBGP over PPPoE, see Sample Topology for eBGP over PPPoE, on page 373.

BNG over Pseudowire Headend

BNG provides subscriber support over Pseudowire Headend (PWHE). PWHE provides L3 connectivity to customer edge nodes through a pseudowire connection. PWHE terminates the L2VPN circuits that exists between the access-provide edge (A-PE) nodes, to a virtual interface, and performs routing on the native IP packet. Each virtual interface can use one or more physical interfaces towards the access cloud to reach customer routers through the A-PE nodes. This feature is supported for PPPoE PTA, PPPoE LAC and IPoE subscribers.

For basic PWHE, the access pseudowire (PW) is terminated on an interface in the Services-PE (S-PE) box. The pseudowire in the access network can be of VC type 4 (tagged), type 5 (raw) and type 11 (inter-working). VC type 4 and VC type 5 pseudowires are represented by pw-ether interfaces. VC type 11 pseudowire is represented by a pw-iw interface. The physical interfaces that the pw-ether or pw-iw interface use is decided through a pin-down list, which is also called as generic-interface-list or a Tx-list. The access P nodes must ensure that the psuedowire traffic is sent to the S-PE box, on only one of the interfaces in the pin-down list. If not, the traffic is dropped on S-PE.

For PWHE with BNG, the subscribers are enabled only on VC type 5 pseudowires (that is, only tagged subscriber traffic is supported), and therefore, the access-interface for the subscribers can only be PWHE sub-interfaces. For sample topology, sample configurations and various deployment models for subscribers on PWHE, see Sample Topology for BNG over Pseudowire Headend, on page 357.

QoS on BNG Pseudowire Headend

Subscriber support over Pseudowire Headend (PWHE) interface was introduced in Cisco IOS XR Software Release 5.2.0. Further support for QoS features for subscribers on PWHE was introduced in Cisco IOS XR Software Release 5.2.2 as follows:

- Support for PPPoE or IPoE subscribers on PWHE sub-interface (with or without SVLAN policy).
- QoS support at different levels:
  - QoS on per-session PPPoE.
  - QoS on multiple PPPoE sessions associated to the same subscriber line, that is shared policy instance (SPI).
  - QoS at pseudowire level.
  - QoS at physical port-level.
- Support for features such as service accounting and pQoS for PWHE subscribers.

- Support for MPLS EXP marking for PWHE subscriber interfaces.

You can configure same SPI instance (with different policy-maps attached) on the sub-interface of PWHE pin-down members as well as on the subscriber interface. In this scenario, the subscriber sessions come up in spite of having the same SPI instance on the pin-down member of PWHE.

For ASR 9000 Enhanced Ethernet Line Card, there are 4 chunks per network processor (NP), and physical interfaces are mapped to a particular NP and chunk. The SE model of this line card (LC) supports 8K subscribers per chunk. To support this, these guidelines must be followed:

- Pin-down members must be distributed so that they are not from the same NP and chunk.
- The **resource-id** option in **service-policy** command must be used to change the chunk mapping of the physical interface.
- The target chunk must not be used by any other interface or sub-interface policy-map.
- The scale is expected to reduce if service accounting is enabled.

Note For BNG PWHE with QoS, an extra 4 bytes per packet get added if service accounting is enabled. This is because of the internal VC label that gets added when the packet enters the ingress LC. This is applicable only for egress direction.

Features Supported for BNG over Pseudowire Headend

These are supported for BNG over PWHE:

- Features such as http-r, Access-Control List (ACL), Accounting, Change of Authorization (CoA) and Lawful-Intercept.
- 64K dual stack and 128K IPv4 subscribers.
- Ambiguous VLANs on PWHE sub-interfaces.
- RFC-3107, for basic PWHE forwarding path from the core to the subscriber direction.
- QoS for the subscribers.
- Other features as applicable for the subscriber.

The supported control protocols for BNG over PWHE are DHCPv4, DHCPv6, IPv6 ND, PPP and PPPoE.

The pw-ether sub-interfaces are also supported in BNG. Ideally, the VC type for the PW can be negotiated as Type 4 or Type 5, for pw-ether interfaces. The pw-ether sub-interfaces are only supported for VC type 5.

These are the supported behavioral models of PWHE for the VC type and the sub-interface:

- According to the standards, the VC type 4 mandates that the SP-VLAN be carried along with the C-VLAN, in the PW. The VC type 5 mandates that the SP-VLAN be removed, and only the C-VLAN be carried in the PW.
- There are implementation differences (mainly in the number of VLANs that are transported in the PW) between Cisco 7600 Series Routers and Cisco ASR 9000 Series Aggregation Services Routers, and Cisco 12000 Series Routers based platforms. However, this does not impact the behavior of A-PE and S-PE.

- Because pw-ether sub-interfaces are supported only for VC type 5, the packet in the PW does not have the SP-VLAN. Therefore, when the subscriber connection enters the S-PE (BNG router), it finds a match with a pw-ether sub-interface VLAN and the C-VLAN in the packet.
- When VC type 4 is configured, it is always matched with the pw-ether main interface. Even if sub-interfaces are configured with VC type 4, they are not used. The system does not restrict the configuration of sub-interfaces.

The hardware support for BNG over PWHE is same as that for the bundle subscriber support. The RSP types supported are RSP-440-SE and RSP-880-SE.

Unsupported Features and Restrictions for BNG over Pseudowire Headend

These are the unsupported features and restrictions for BNG over PW HE feature:

- Subscribers on VC type-4 and VC type-11 pseudowires are not supported (that is, untagged subscribers cannot be terminated on a BNG PWHE interface and they are restricted in CLI on the main pw-ether interface).
- Egress subscriber Lawful-Intercept is not supported.
- Multicast for PPPoE is not supported.
- SPAN is not supported.
- Cluster is not supported.
- IPoE L3 connected subscribers are not supported.
- Because subscribers on PWHE are based out of RP, linecard (LC) subscribers are not supported.
- Because satellite is not supported on PWHE, it is not supported on PWHE over BNG too.
- IPv4 fragmentation is not supported.

The support for QoS on BNG PWHE is subjected to these restrictions:

- PWHE subscribers are supported only in Co-existence disabled mode of line card (LC).
- ATM overhead accounting is not supported.
- Because multicast is not supported on PWHE subscriber, IGMP shaper co-relation is not supported.

PPPoE LAC Subscriber Over PWHE

The PPPoE LAC session over Pseudowire Headend (PWHE) feature enables LAC session to be established on PWHE interface. The PWHE technology allows termination of Access Pseudowire into a Layer 3 (VRF or global) domain or into a Layer 2 domain. PWHE infrastructure enables an easy and scalable mechanism for tunneling or backhauling traffic into a common IP, MPLS, or L2 network.

Supported Features

- Lawful Intercept (LI)
- uRPF

- Subscriber Control Plane Policing (CoPP)
- HTTP-Redirect (HTTPr)

Restrictions

- Routing protocols cannot be run on the subscriber interfaces
- L2TP is not supported for IP subscribers
- L2TP limitations are applicable with respect to sequencing, fragmentation, and checksums as applied to bundle-based LAC sessions
- L2TP imposition is not supported on A9K-SIP-700 Line Cards or Cisco ASR 9000 Series SPA Interface Processor-700
- VC type 4 and 11 are not supported for hosting subscribers
- For ingress L2TP packets, the negotiated UDP destination port is 1701 and the source port is defined by the LNS

Unsupported Features

- Routed subscriber session is not supported
- Multicast is not supported for PPPoE sessions over PWHE
- Cluster, satellite, and geo-redundancy are not supported
- SPAN and egress LI are not supported
- ACL is not applicable on BNG sessions, as the incoming and outgoing traffic flow through MPLS routing
- Quality of Service (QoS)
- Layer 2 Tunnel Protocol Version 3 (L2tpv3)

Additional References

These sections provide references related to implementing PPP, PPPoE, L2TP, and DHCP.

RFCs

Standard/RFC - PPP	Title
RFC-1332	The PPP Internet Protocol Control Protocol (IPCP)
RFC-1570	PPP LCP Extensions
RFC-1661	The Point-to-Point Protocol (PPP)
RFC-1994	PPP Challenge Handshake Authentication Protocol (CHAP)

Standard/RFC - PPPoE	Title
RFC-2516	A Method for Transmitting PPP Over Ethernet (PPPoE)
RFC-4679	DSL Forum Vendor-Specific RADIUS Attributes

Standard/RFC - L2TP	Title
RFC-2661	Layer two tunneling protocol "L2TP"

MIBs

MIBs	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 6

Deploying the Quality of Service (QoS)

The Quality of Service (QoS) feature ensures that traffic to and from priority applications gets preference in using network resources. QoS actions are defined by service-policies that are deployed using policy-maps. During the QoS process, packets are encapsulated with QoS information. The encapsulation is monitored and accounted by the QoS accounting function.

Parameterized QoS (PQoS) is another form of QoS in which the traffic priority is based on the characteristic of the data being carried by the traffic.

BNG supports merging of multiple QoS policy-maps applied through multiple dynamic templates and implementing them on a single subscriber.

Table 8: Feature History for Configuring Subscriber Features

Release	Modification
Release 4.2.0	Initial release of this document.

This chapter explains deploying QoS, and covers the following topics:

Quality of Service Overview

Quality of Service (QoS) is the technique of prioritizing network traffic for time-sensitive and mission-critical applications such as VoIP services, live streaming of videos, priority accesses to database, and so on. Functions that QoS provides ensure that such applications receive sufficient bandwidth at low latency, with reduced data loss.

QoS functions perform preferential forwarding of high-priority packets. To do so, the packet is identified, classified, and then prioritized on all routers along the data-forwarding path throughout the network. As a result, priority applications get the resources they require, while other applications access the network, simultaneously.

QoS functions provide service providers cost benefits by enabling them to use existing resources efficiently and ensure the required level of service without reactively expanding, or over-provisioning their networks. QoS also improves customer experience when they get reliable services from a variety of network applications.

It is ideal to deploy QoS on BNG because BNG is present at the edge router, and subscriber directly connects to it. One of the unique features of BNG is QoS accounting. This feature enables BNG to collect and report QoS encapsulation information to the RADIUS server. For details, see QoS Accounting, on page 200.

The deployment of QoS involves three components:

- Class-map — Classifies different forms of traffic, like video, data, VOIP and so on, based on matching rules.
- Policy-map — Defines the QoS actions to be applied to the classified traffic. It references the classes previously defined in the class-map. These policy-maps are also called QoS maps. The actions defined in the policy-map perform traffic prioritization and bandwidth allocation.
- Service policy — Associates a previously defined policy-map with a attachment point and direction, on BNG. The attachment points are listed in the section QoS Attachment Points, on page 214.The two directions possible for a policy is input and output. The policy direction is relative to the attachment point.

BNG supports two-level hierarchical policy (parent policy and child policy) for deploying QoS. Based on the preference of service provider, the QoS policies are defined and applied on BNG in these ways:

- Define and apply the QoS policy from CLI. See, Configuring Service-policy and Applying Subscriber Settings Through Dynamic Template, on page 178.
- Define the QoS policy in CLI, but apply it from RADIUS. See, Configuring Service-policy and Applying Subscriber Settings Through RADIUS, on page 176.
- Define and apply the QoS policy from RADIUS. It is also called Parameterized QoS, on page 180.

Restriction

- If the subscriber ingress or egress QoS includes policing, shaping, bandwidth, or WRED actions, it is recommended that only active:standby bundle interfaces be used. Load-sharing should be avoided.
- Users can configure only 7 class-maps (both ingress and egress, excluding the class-default map) to achieve a higher scale configuration (say, 32,000 sessions) per node processor.

Configuring Service-policy and Applying Subscriber Settings Through RADIUS

Perform this task to deploy the QoS policy using CLI commands. In this task, subscriber settings are applied from the RADIUS server.

SUMMARY STEPS

1. **configure**
2. **policy-map type qos** *q_in*
3. **class class-default**

4. **service-policy** *q_child_in*
5. **policy-map type qos** *q_out*
6. **class class-default**
7. **service-policy** *q_child_out*
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type qos** *q_in* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type qos q_in`	Configures the policy-map for the type qos.
Step 3	**class class-default** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	Configures or modifies the parent class-default class. **Note** You can configure only the class-default class in a parent policy. Do not configure any other traffic class.
Step 4	**service-policy** *q_child_in* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy q_child_in`	Applies a bottom-level policy to the top-level class-default class.
Step 5	**policy-map type qos** *q_out* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type qos q_out`	Configures the policy-map for the type qos.
Step 6	**class class-default** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	Configures or modifies the parent class-default class. **Note** You can configure only the class-default class in a parent policy. Do not configure any other traffic class.
Step 7	**service-policy** *q_child_out* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy q_child_out`	Applies a bottom-level policy to the top-level class-default class.
Step 8	**commit**	

Configuring Subscriber Policy through CLI and Applying through RADIUS: Examples

```
configure
policy-map type qos q_in
class class-default
end

\\the following procedure is ran in RADIUS
     Service-Type = Outbound-User
     Cisco-avpair = "ipv4:ipv4-mtu=750",
     Cisco-avpair = "ipv4:ipv4-unnumbered=Loopback0",
     Cisco-avpair = "subscriber:sub-qos-policy-in=q_in",
     Cisco-avpair = "subscriber:sub-qos-policy-out=q_out",
     Idle-Timeout = 1000,
     Session-Timeout = 5000
```

Configuring Service-policy and Applying Subscriber Settings Through Dynamic Template

Perform this task to deploy the QoS policy using CLI commands. In this task, subscriber settings are applied using a dynamic template.

SUMMARY STEPS

1. **configure**
2. **policy-map type qos** *q_in*
3. **class class-default**
4. **service-policy** *q_child_in*
5. **policy-map type qos** *q_out*
6. **class class-default**
7. **service-policy** *q_child_out*
8. **dynamic-template type ppp** *dynamic_config*
9. **service-policy input** *q_in*
10. **service-policy output** *q_out*
11. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type qos** *q_in* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type qos q_in`	Configures the policy-map in the input direction.
Step 3	**class class-default**	Configures or modifies the parent class-default class.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	**Note** You can configure only the class-default class in a parent policy. Do not configure any other traffic class.
Step 4	**service-policy** *q_child_in* **Example:** `RP/0/RSP0/CPU0:router(config)# service-policy q_child_in`	Configures the service policy for the input direction. **Note** The q_in and q_out policy maps are parent policy maps.
Step 5	**policy-map type qos** *q_out* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type qos q_out`	Configures the policy-map for the output direction. **Note** The q_in and q_out policy maps are parent policy maps.
Step 6	**class class-default** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	Configures or modifies the parent class-default class. **Note** You can configure only the class-default class in a parent policy. Do not configure any other traffic class.
Step 7	**service-policy** *q_child_out* **Example:** `RP/0/RSP0/CPU0:router(config)# service-policy q_child_out`	Applies a bottom-level policy to the top-level class-default class. **Note** The q_in and q_out policy maps are parent policy maps.
Step 8	**dynamic-template type ppp** *dynamic_config* **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template type ppp dynamic_config`	Configures dynamic-template of the type ppp and applies the configuration through dynamic-template.
Step 9	**service-policy input** *q_in* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy input q_in`	Configures the service-policy in the input direction.
Step 10	**service-policy output** *q_out* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy input q_out`	Configures the service-policy in the output direction.
Step 11	**commit**	

Configuring Subscriber Policy through CLI and Applying to Subscriber through Dynamic-Template: Examples

```
configure
policy-map type qos q_in  // policy-map input direction
class class-default
end

configure
policy-map type qos q_out  // policy-map output direction
class class-default
end

// applying configuration through dynamic-template
configure
dynamic-template type ppp dynamic_policy
service-policy input q_in
service-policy output q_out
end
```

Parameterized QoS

Parameterized Quality of Service (PQoS) guarantees reliable performance of a network application by reserving for it the required network bandwidth. In this case, the prioritization is based on the type of data being carried by the packet.

In the standard QoS, the importance of a packet is based on the priority level that is defined for it. It is possible that in once case a video packet and an asynchronous data transfer packet have the same priority level defined. In such a case, the router gives equal importance to both packets. As a result, because of bandwidth conflict, there can be video degradation.

On the other hand, in PQoS, packet importance is based on the characteristics or parameters of the data that is carried by the packet. For example, it is possible to have PQoS provide dedicated bandwidth for video packets. Even at times when heavy loads of asynchronous data traffic are introduced into the network, PQoS guarantees that video packets have priority over other data streams that do not require real-time streaming.

Parameterized QoS has the ability to define, modify, or delete QoS policy-map based Vendor Specific Attributes (VSAs). VSAs are downloaded through the RADIUS server. The attributes from the parameterized QoS policies are filtered and passed on to the policy object library; the latter parses and translates them into policy objects. The VSAs define a two-level hierarchical policy to be applied on the subscriber session. The format of the QoS VSAs is:

```
AVPair: qos-policy-in=add-class(sub,<parent-class, child-class>,<action-list>)
AVPair: qos-policy-out=add-class(sub,<parent-class, child-class>,<action-list>)
AVPair: qos-policy-in=remove-class(sub,<parent-class, child-class>)
AVPair: qos-policy-out=remove-class(sub,<parent-class, child-class>)
```

where:

- “sub”, is a constant string, signifies that the current policy on the subscriber is to be modified
- <class-list> gives the hierarchy of the class to be added or removed (i.e. parent-class, child-class)

- <action-list> gives the QoS actions to be applied under the class being added

For more information about QoS parameters and its syntax, see *Parameterized QoS Syntax* in the Configuring Parameterized QoS Policy Through RADIUS, on page 186.

When a parameterized QoS policy for a subscriber is downloaded from the RADIUS server for the first time, the VSAs are used to build the policy from scratch. After the policy is applied on the subscriber, any new or modified VSAs downloaded for that subscriber from the RADIUS server automatically modifies the already applied policy.

For deploying a Parameterized QoS policy from the RADIUS server, see Configuring Parameterized QoS Policy Through RADIUS, on page 186.

Using Change of Authorization (CoA), it is possible to update the service-policy by modifying the class-maps that were previously configured by the parameterized QoS. Modifying can involve removing existing classes, or adding new classes. To make updates to the service-policy, see Modifying Service Policy through CoA , on page 188.

Parameterized QoS Syntax

Parameterized QoS Syntax

QoS Action Parameter	Qualifiers	Commands
Shape	QoS Action	shape(<rate-in-kbps>)
	CLI Equivalent	shape average <shape-rate> <kbps>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default),shape(14700))
Shape in percentage	QoS Action	Shape-rpct(<rate-in-pct>)
	CLI Equivalent	shape average percent < rate-in-pct >
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default),shape-pct(25))
Police (Variant 1)	QoS Action	police(<conform-rate-in-kbps>, <conform-burst-in-kBytes>, <exceed-rate-in-kbps>, <exceed-burst-in-kbytes>, <conform-action>, <exceed-action>, <violate-action>)

QoS Action Parameter	Qualifiers	Commands
	CLI Equivalent	police rate <conform-rate> <kbps> burst <conform-burst> <kbps> peak-rate <exceed-rate> exceed-burst <exceed-burst> conform-action <action> exceed-action <action> violate-action <action>
	RADIUS Equivalent - Example	qos-policy-in:add-class(sub,(class-default, voip),police(2000,2000, 4000, 4000,transmit, set-ipprec(< precedence>), drop))
Police (Variant 2)	QoS Action	Police (<conform-rate-in-kbps>)
	CLI Equivalent	police rate <kbps>
	RADIUS Equivalent - Example	qos-policy-in:add-class(sub,(class-default, voip), police(200000))
Police in percentage (Variant 1)	QoS Action	police-rpct(<conform-rate-in-pct>, <conform-burst-in-us>, <exceed-rate-in-pct>, <exceed-burst-in-us>, <conform-action>, <exceed-action>, <violate-action>)
	CLI Equivalent	police rate percentage <pct> burst <conform-burst> < us> peak-rate percentage<pct> exceedburst <exceed-burst> conform-action <action> exceed-action <action> violate-action <action>
	RADIUS Equivalent - Example	qos-policy-in:add-class(sub,(class-default, voip),police-rpct(20,20, 40, 40,transmit, set-ipprec(< precedence>), drop))
Police in percentage (Variant 2)	QoS Action	Police-rpct(<conform-rate-in-pct>

QoS Action Parameter	Qualifiers	Commands
	CLI Equivalent	police rate percentage <pct>
	RADIUS Equivalent - Example	qos-policy-in:add-class(sub,(class-default, voip), police-rpct(20))
Set IP Precedence	QoS Action	set-ip-prec(<precedence>)
	CLI Equivalent	set precedence <precedence>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip), set-ip-prec(5))
Set CoS	QoS Action	set-cos(<cos-val>)
	CLI Equivalent	set cos <cos-val>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip), set-cos(5))
Minimum Bandwidth	QoS Action	bw-abs(<bw-in-kbps>)
	CLI Equivalent	bandwidth <bw-in-kbps>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,video),bw-abs(2000))
Minimum bandwidth percentage	QoS Action	bw-pct(<bw-in-pct>)
	CLI Equivalent	bandwidth percent <pct>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,video),bw-abs(2000))
Bandwidth Remaining Percentage	QoS Action	bw-rpct(<pct>)
	CLI Equivalent	bandwidth remaining percent <pct>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip),bw-rpct(33))
Set IP DSCP	QoS Action	set-ip-dscp(<dscp-val>)

QoS Action Parameter	Qualifiers	Commands
	CLI Equivalent	Set dscp <dscp-val>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip), set-ip-dscp(46))
Queue Limit in packets	QoS Action	queue-limit(<qlimit-in-packets>)
	CLI Equivalent	queue-limit <val> < packets>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip),queue-limit(64))
Queue Limit in us	QoS Action	queue-limit-us(<qlimit-in-us>)
	CLI Equivalent	queue-limit <val> <us>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip),queue-limit-us(240))
DSCP based WRED	QoS Action	random-detect-dscp(<dscp>, <min-threshold>, <max-threshold>, <probability>)
	CLI Equivalent	random-detect dscp <dscp-val> < Min-thresh> <Kbytes> <max-thresh> <Kbytes> probability < probability-val>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip), random-detect-dscp (24, 25000, 35000))
Precedence based WRED	QoS Action	random-detect-prec (<precedence>, <min-threshold>, <max-threshold>, <probability>)
	CLI Equivalent	random-detect precedence <prec-val> < Min-thresh> <Kbytes> <max-thresh> <Kbytes> probability < probability-val>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip), random-detect- (24, 25000, 35000))
Set qos group	QoS Action	set-qos-grp(<group-val>)
	CLI Equivalent	set qos-group <qos-group-val>

QoS Action Parameter	Qualifiers	Commands
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip), set-qos-grp (24))
Priority Level	QoS Action	pri-level(<priority-level>)
	CLI Equivalent	priority level <priority-level>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default, voip), pri_level(1))
Set discard class	QoS Action	set-dclass(<discard-class-val>)
	CLI Equivalent	set discard-class <discard-class-val>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip), set-dclass (4))
Set MPLS exp topmost bit	QoS Action	set-mpls-exp-topmost (<mpls-exp- topmost-val>)
	CLI Equivalent	set mpls experimental topmost <mpls-exp- topmost-val>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip), set-mpls-exp-topmost (4))
Set MPLS exp imposition bit	QoS Action	set-mpls-exp- imposition (<mpls-exp-imposition-val>)
	CLI Equivalent	set mpls experimental imposition <mpls-exp- imposition-val>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip), set-mpls-exp-imposition (4))
Set Tunnel precedence	QoS Action	set-tunnel-prec(<prec-val>)
	CLI Equivalent	set precedence tunnel <precedence-val>
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip), set-tunnel-prec(4))
Set Tunnel DSCP	QoS Action	set-tunnel-dscp (<dscp-val>)
	CLI Equivalent	set dscp tunnel <dscp-val>

QoS Action Parameter	Qualifiers	Commands
	RADIUS Equivalent - Example	qos-policy-out:add-class(sub,(class-default,voip), set-tunnel-dscp(4))

Configuring Parameterized QoS Policy Through RADIUS

Perform this task to deploy parameterized QoS policy and apply subscriber settings through the RADIUS server. These steps are performed on the RADIUS server for each subscriber.

Note

- Parameterized QoS configuration through the RADIUS server is applicable only for user-profiles; not for service-profiles.
- In parameterized QoS configuration, the policy-map is not defined on the CLI. It is dynamically created based on the configuration passed through RADIUS. This procedure applies to the RADIUS server as part of RADIUS user configurations. The policy-map results are applied to the subscriber when that user profile is downloaded after executing a control policy authentication or authorization action. The class-map must be configured through CLI. For this task, the *classes voice_in*, *video_in*, *data_in*, *video_out*, *voice_out*, and *data_out* are configured separately.

SUMMARY STEPS

1. **Cisco-AVPair** = *"ip:qos-policy-in=add-class(sub, (class-default),police(2000))"*
2. **Cisco-AVPair**+= *"ip:qos-policy-in=add-class(sub, (class-default,voice_in), pri-level(1), police(256))"*
3. **Cisco-AVPair**+= *ip:qos-policy-in=add-class(sub, (class-default,video_in), pri-level(2), police(1000))"*
4. **Cisco-AVPair** += *"ip:qos-policy-in=add-class(sub, (class-default,data_in), set-qos-grp(4))"*
5. **Cisco-AVPair** += *"ip:qos-policy-in=add-class(sub, (class-default,class-default), set-qos-grp(7))"*
6. **Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default), shape(4000))"*
7. **Cisco-AVPair** += *"ip:qos-policy-out=add-class(sub, (class-default,voice_out), pri-level(1),queue-limit-us(10000))"*
8. **Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default,video_out),queue-limit-us(30000), shape(2000))"*
9. **Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default,data_out), bw-rpct(20))"*
10. **Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default,class-default))"*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**Cisco-AVPair** = *"ip:qos-policy-in=add-class(sub, (class-default),police(2000))"* **Example:** `Cisco-AVPair = "ip:qos-policy-in=add-class(sub, (class-default),police(2000))"`	Configures the cisco-avpair class-map in input direction for police action parameter.

	Command or Action	Purpose
Step 2	**Cisco-AVPair**+= *"ip:qos-policy-in=add-class(sub, (class-default,voice_in), pri-level(1), police(256))"* **Example:** `Cisco-AVPair =  "ip:qos-policy-in=add-class(sub, (class-default,voice_in), pri-level(1), police(256))"`	Configures the cisco-avpair class-map in input direction for the police action parameter.
Step 3	**Cisco-AVPair**+= *ip:qos-policy-in=add-class(sub, (class-default,video_in), pri-level(2), police(1000))"* **Example:** `Cisco-AVPair = ip:qos-policy-in=add-class(sub, (class-default,video_in), pri-level(2), police(1000))"`	Configures the cisco-avpair class-map in input direction for the police action parameter.
Step 4	**Cisco-AVPair** += *"ip:qos-policy-in=add-class(sub, (class-default,data_in), set-qos-grp(4))"* **Example:** `Cisco-AVPair = "ip:qos-policy-in=add-class(sub, (class-default,data_in), set-qos-grp(4))"`	Configures the cisco-avpair class-map in input direction for the police action parameter.
Step 5	**Cisco-AVPair** += *"ip:qos-policy-in=add-class(sub, (class-default,class-default), set-qos-grp(7))"* **Example:** `Cisco-AVPair = "ip:qos-policy-in=add-class(sub, (class-default,class-default), set-qos-grp(7))"`	Configures the cisco-avpair class-map in input direction for the set qos action parameter.
Step 6	**Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default), shape(4000))"* **Example:** `Cisco-AVPair = "ip:qos-policy-out=add-class(sub, (class-default), shape(4000))"`	Configures the cisco-avpair class-map in output direction for the shape action parameter.
Step 7	**Cisco-AVPair** += *"ip:qos-policy-out=add-class(sub, (class-default,voice_out), pri-level(1),queue-limit-us(10000))"* **Example:** `Cisco-AVPair = "ip:qos-policy-out=add-class(sub, (class-default,voice_out), pri-level(1),queue-limit-us(10000))"`	Configures the cisco-avpair class-map in output direction for the queue-limit-us action parameter.
Step 8	**Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default,video_out),queue-limit-us(30000), shape(2000))"*	Configures the cisco-avpair class-map in output direction for the queue-limit-us and the shape action parameters.

	Command or Action	Purpose
	Example: `Cisco-AVPair = "ip:qos-policy-out=add-class(sub, (class-default,video_out),queue-limit-us(30000), shape(2000))"`	
Step 9	**Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default,data_out), bw-rpct(20))"* **Example:** `Cisco-AVPair = "ip:qos-policy-out=add-class(sub, (class-default,data_out), bw-rpct(20))"`	Configures the cisco-avpair class-map in output direction for the bandwidth action parameter.
Step 10	**Cisco-AVPair**+= *"ip:qos-policy-out=add-class(sub, (class-default,class-default))"* **Example:** `Cisco-AVPair = "ip:qos-policy-out=add-class(sub, (class-default,class-default))"`	Configures the cisco-avpair class-map in output direction for the class action parameter. **Note** For the complete list of QoS action parameters that can be configured and applied through RADIUS, see *Parameterized QoS Syntax* section in Parameterized QoS Syntax, on page 181.

Configuring Parameterized Subscriber Policy Defined and Applied through RADIUS: An example

```
Cisco-AVPair = "ip:qos-policy-in=add-class(sub, (class-default),police(2000))"
Cisco-AVPair += "ip:qos-policy-in=add-class(sub, (class-default,voice_in), pri-level(1),
police(256))"
Cisco-AVPair += "ip:qos-policy-in=add-class(sub, (class-default,video_in), pri-level(2),
police(1000))"
Cisco-AVPair += "ip:qos-policy-in=add-class(sub, (class-default,data_in), set-qos-grp(4))"

Cisco-AVPair += "ip:qos-policy-in=add-class(sub, (class-default,class-default),
set-qos-grp(7))"
Cisco-AVPair += "ip:qos-policy-out=add-class(sub, (class-default), shape(4000))"
Cisco-AVPair += "ip:qos-policy-out=add-class(sub, (class-default,voice_out),
pri-level(1),queue-limit-us(10000))"
Cisco-AVPair += "ip:qos-policy-out=add-class(sub,
(class-default,video_out),queue-limit-us(30000), shape(2000))"
Cisco-AVPair += "ip:qos-policy-out=add-class(sub, (class-default,data_out), bw-rpct(20))"

Cisco-AVPair += "ip:qos-policy-out=add-class(sub, (class-default,class-default))"
```

Modifying Service Policy through CoA

Perform this task to modify service-policy through CoA.

Note The Web Portal or Radius server that supports CoA should be configured to generate a CoA request with Cisco VSA corresponding to the steps in this task.

SUMMARY STEPS

1. **qos-policy-out** *remove-class(sub, (class-default, voip))*
2. **qos-policy-out** *add-class(sub, (class-default, video), bw-rpct(50), pri-level(2))*
3. **qos-policy-out** *add-class(sub, (class-default, data), shape(400),set-ip-prec(1))*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**qos-policy-out** *remove-class(sub, (class-default, voip))* **Example:** `qos-policy-out=remove-class(sub, (class-default, voip))`	Removes the class map, where voip is the class to be removed from a previously configured parameterized QoS for a subscriber.
Step 2	**qos-policy-out** *add-class(sub, (class-default, video), bw-rpct(50), pri-level(2))* **Example:** `qos-policy-out=add-class(sub, (class-default, video), bw-rpct(50), pri-level(2))`	Adds a class map, where video is the class to be added to a previously configured parameterized QoS for a subscriber.
Step 3	**qos-policy-out** *add-class(sub, (class-default, data), shape(400),set-ip-prec(1))* **Example:** `qos-policy-out=add-class(sub, (class-default, data), shape(400),set-ip-prec(1))`	Configures the qos-policy-out for shape, set ip precedence parameters.

Modifying Service Policy through CoA : Examples

```
//Policy-map configuration before CoA
policy-map __sub_5e311c4f_child1
 class voip
  priority level 1
  police rate 10000 kbps burst 8 kbytes
   !
!
class video
  priority level 1
  police rate 10000 kbps burst 16 kbytes
   !
!
class data
  shape average 80000 kbps
!
class class-default
!
end-policy-map
!
policy-map __sub_5e311c4f
 class class-default
  service-policy __sub_5e311c4f_child1
```

```
  shape average 100000 kbps
!
end-policy-map
!

//Modifying Service Policy through CoA
qos-policy-out=remove-class(sub, (class-default, voip))
qos-policy-out=add-class(sub, (class-default, video), bw-rpct(50), pri-level(2))
qos-policy-out=add-class(sub, (class-default, data), shape(400),set-ip-prec(1))

//Policy-map configuration after CoA looks like:
 policy-map __sub_ffffffec1a37f_child1
class video
priority level 2
  bandwidth percent 50
  police rate 10000 kbps burst 16 kbytes
  !
!
class data
  shape average 400 kbps
  set precedence 1
!
class class-default
!
end-policy-map
!
policy-map __sub_ffffffec1a37f
class class-default
  service-policy __sub_ffffffec1a37f_child1
  shape average 100000 kbps
!
end-policy-map
!
```

Parameterized QoS for Line Card Subscribers

From Cisco IOS XR Release 5.3.2 and later, parameterized QoS (PQoS) as auto-service is supported for LC subscribers, along with RP subscribers. For PQoS as auto-service, all the PQoS attributes are defined as VSAs in the service profile, and activated as auto-service from the user profile. The regular mode of PQoS, where the attributes are defined in user profile and activated by a service logon CoA request, is not supported for LC subscribers. Whereas, RP subscribers support both modes of PQoS.

In user profile-based PQoS, the entire set of Cisco-AVPairs needs to be downloaded every time a new session comes up. Whereas, for PQoS as auto-service, the attributes need to be downloaded only for the first session. If the same service is to be activated for the next session, the attributes that were downloaded earlier for the previous session can be used from the BNG router itself. This reduces the processing time considerably and provides more flexibility in activating and deactivating a service.

To deactivate a PQoS service, use the service-logoff request irrespective of the way it was activated. To modify the PQoS feature per subscriber session, send a multi-action CoA request with a deactivation command for the active service (**cisco-avpair += "subscriber:sd=<old-service>"**) and an activation command for the new service (**cisco-avpair += "subscriber:sa=<new-service>"**). To modify the service definitions which are currently used by the session, send the service update CoA request with new parameters.

Configuring Parameterized QoS as Auto-service

Configuration Guidelines

- For each service in the user profile, there must be a corresponding **Method-List** specified. Else, the BNG router considers that the service profile is defined locally.
- Once you download pqos as auto-service from the RADIUS server, the only way to change the service definition in the router, is through a CoA service-update request.
- The CoA account status query might not reply the **echo-strings** for the service.
- While a session starts, the user might want to apply default QoS service apart from the PQoS service. In such cases, ensure that the default QoS profile is applied as service template and activated after the authentication action. This avoids multiple instances of apply and undo apply during session bring-up, thereby providing good bring-up calls-per-second (CPS). It also avoids unnecessary feature installation for access-rejected users as well.

Configuration of PQoS as Auto-service: Example

This example shows a sample user profile and service profile, to activate the services 1_Mbps_IN and 1_Mbps_OUT as auto-service:

User Profile:

```
BNGuser1@bngtm.com Cleartext-Password := cisco
  service-Type=Framed-User,
  Cisco-AVPair += "echo-string-1=1_Mbps_IN",
  Cisco-AVPair += "echo-string-2=1_Mbps_OUT",
  Framed-Filter-Id = ACL_VOZ_CONTROL_IN.in,
  Cisco-avpair += "subscriber:sa=1_Mbps_IN”,
  Cisco-AVPair += "Method-List=default"
  Cisco-avpair += "subscriber:sa=1_Mbps_OUT",
  Cisco-AVPair += "Method-List=default
```

Service Profile:

```
1_Mbps_IN  Cleartext-Password := "cisco"
       Cisco-AVPair += "ip:qos-policy-in=add-class(sub,(class-default),police(1085))",
       Cisco-AVPair +=
"ip:qos-policy-in=add-class(sub,(class-default,BROADBAND_VOZ),police(512,transmit,drop),set-mpls-exp-imposition(5))",

       Cisco-AVPair +=
"ip:qos-policy-in=add-class(sub,(class-default,BROADBAND_CRITICOS),set-mpls-exp-imposition(1))",

       Cisco-AVPair +=
"ip:qos-policy-in=add-class(sub,(class-default,BROADBAND_BUSINESS),set-mpls-exp-imposition(1))",

       Cisco-AVPair +=
"ip:qos-policy-in=add-class(sub,(class-default,class-default),set-mpls-exp-imposition(0),set-ip-dscp(0))”

1_Mbps_OUT  Cleartext-Password := "cisco"
       Cisco-AVPair += "ip:qos-policy-out=add-class(sub,(class-default),shape(1064))",
       Cisco-AVPair +=
"ip:qos-policy-out=add-class(sub,(class-default,BROADBAND_VOZ),police(512,transmit,drop),pri-level(1),set-cos(5))",

       Cisco-AVPair +=
"ip:qos-policy-out=add-class(sub,(class-default,BROADBAND_CRITICOS),bw-rpct(50),set-cos(1))",
```

```
       Cisco-AVPair +=
"ip:qos-policy-out=add-class(sub,(class-default,BROADBAND_BUSINESS),bw-rpct(35),set-cos(1))",

       Cisco-AVPair +=
"ip:qos-policy-out=add-class(sub,(class-default,class-default),bw-rpct(15),set-cos(0))"
```

Single CoA Request: Example

This example shows a single CoA request to activate a PQoS service:

```
echo "Acct-Session-Id=08000001,Cisco-avpair+='subscriber:sa=pQOS_SVC_1MIN',
Cisco- AVPair+='Method-List=default'" | /usr/local/bin/radclient -x 6.6.6.18:1500 coa cisco
 -r 1
Sending CoA-Request of id 134 to 6.6.6.18 port 1500 Acct-Session-Id = "08000001"
Cisco-AVPair += "subscriber:sa=pQOS_SVC_1MIN"
Cisco-AVPair += "Method-List=default"
rad_recv: CoA-ACK packet from host 6.6.6.18 port 1500, id=134, length=50
Cisco-AVPair = "sa=pQOS_SVC_1MIN"
```

This example shows a single CoA request to deactivate a PQoS service:

```
echo "Acct-Session-Id=08000001,Cisco-avpair+='subscriber:sd=pQOS_SVC_1MIN',
Cisco- AVPair+='Method-List=default'" | /usr/local/bin/radclient -x 6.6.6.18:1500 coa cisco
 -r 1
Sending CoA-Request of id 21 to 6.6.6.18 port 1500 Acct-Session-Id = "08000001"
Cisco-AVPair += "subscriber:sd=pQOS_SVC_1MIN"
Cisco-AVPair += "Method-List=default"
rad_recv: CoA-ACK packet from host 6.6.6.18 port 1500, id=21, length=50
Cisco-AVPair = "sd=pQOS_SVC_1MIN"
```

Multi-action CoA Request: Example

This example shows a sample multi-action CoA request used to deactivate the service, pQOS_SVC_1MOUT and to activate the service, pQOS_SVC_2MOUT. It also updates the corresponding echo-string in single CoA request:

```
echo "Acct-Session-Id=080043e5,Cisco-Avpair+='subscriber:sd=pQOS_SVC_1MOUT',cisco-
avpair+='Method-List=default',Cisco-AVPair+='echo-string-2=2_Mbps_OUT',
Cisco- avpair+='subscriber:sa=pQOS_SVC_2MOUT',cisco-avpair+='Method-List=default'" |
/usr/local/bin/radclient -x 6.6.6.18:1500 coa cisco

Sending CoA-Request of id 77 to 6.6.6.18 port 1500 Acct-Session-Id = "080043e5"

Cisco-AVPair += "subscriber:sd=pQOS_SVC_1MOUT" Cisco-AVPair += "Method-List=default"
Cisco-AVPair += "echo-string-1=1_Mbps_OUT"
Cisco-AVPair += "echo-string-2=2_Mbps_OUT" Cisco-AVPair += "subscriber:sa=pQOS_SVC_2MOUT"
Cisco-AVPair += "Method-List=default"
rad_recv: CoA-ACK packet from host 6.6.6.18 port 1500, id=77, length=80 Cisco-AVPair =
"sd=pQOS_SVC_1MOUT"  Cisco-AVPair = "sa=pQOS_SVC_2MOUT"
```

Service-update CoA Request: Example

This example shows a sample service-update CoA request to modify the parameters of a policy-map that is active on the BNG router:

```
echo "cisco-avpair+='subscriber:command=service-update',Cisco-
avpair+='subscriber:service-name=pQOS_SVC_1MIN',Cisco-AVPair+='Method- List=default',
Cisco-AVPair+='ip:qos-policy-in=add-class(sub,(class-
default),police(1085))',Cisco-AVPair+='ip:qos-policy-in=add-class(sub,
(class- default,BROADBAND_VOZ),police(512,transmit,drop),set-mpls-exp-imposition(4))',
Cisco- AVPair+='ip:qos-policy-in=add-class(sub,(class-default,BROADBAND_CRITICOS),set-mpls-
 exp-imposition(2))',
Cisco-AVPair+='ip:qos-policy-in=add-class(sub,(class-
default,BROADBAND_BUSINESS),set-mpls-exp-imposition(2))',
Cisco-AVPair+='ip:qos-policy-
in=add-class(sub,(class-default,class-default),set-mpls-exp-imposition(0),set-ip- dscp(0))'"
 | /usr/local/bin/radclient -x 6.6.6.18:1500 coa cisco

Sending CoA-Request of id 173 to 6.6.6.18 port 1500 Cisco-AVPair +=
"subscriber:command=service-update"

Cisco-AVPair += "subscriber:service-name=pQOS_SVC_1MIN" Cisco-AVPair += "Method-List=default"
Cisco-AVPair += "ip:qos-policy-in=add-class(sub,(class-default),police(1085))"
Cisco-AVPair += "ip:qos-policy-in=add-class(sub,(class-
default,BROADBAND_VOZ),police(512,transmit,drop),set-mpls-exp-imposition(5))"
Cisco-AVPair += "ip:qos-policy-in=add-class(sub,(class-
default,BROADBAND_CRITICOS),set-mpls-exp-imposition(1))"
Cisco-AVPair += "ip:qos-policy-in=add-class(sub,(class-
default,BROADBAND_BUSINESS),set-mpls-exp-imposition(1))"
Cisco-AVPair += "ip:qos-policy-in=add-class(sub,(class-default,class-
default),set-mpls-exp-imposition(0),set-ip-dscp(1))"

rad_recv: CoA-ACK packet from host 6.6.6.18 port 1500, id=173, length=20
```

Verifying PQoS Configuration

You can use these show commands to verify the PQoS configuration:

- Verify if all the policy parameters, like policer and shaper values received from the RADIUS server, are applied on the interface:

```
router# show policy-map applied interface GigabitEthernet0/0/0/0.1.pppoe4

Input policy-map applied to GigabitEthernet0/0/0/0.1.pppoe4:

  policy-map __sub_655b501d
   class class-default
    service-policy __sub_655b501d_child1
    police rate 2085 kbps
    !
   !

Child policy-map(s) of policy-map __sub_655b501d:

  policy-map __sub_655b501d_child1
   class BROADBAND_VOZ
    police rate 512 kbps
     conform-action transmit
     exceed-action drop
    !
    set mpls experimental imposition 5
   !
   class BROADBAND_CRITICOS
    set mpls experimental imposition 1
   !
   class BROADBAND_BUSINESS
```

```
    set mpls experimental imposition 1
   !
   class class-default
    set mpls experimental imposition 0
    set dscp 0
   !
   end-policy-map
  !
```

- Verify the applied service(s) for the session:

```
router# show subscriber session all detail internal

Interface:                GigabitEthernet0/0/0/0.1.pppoe4
- - -
- - -
Policy Executed:

  event Session-Start match-first [at Wed May 13 10:40:43 2015]
    class type control subscriber PPP_CM do-until-success [Succeeded]
      10 activate dynamic-template PTA_TEMPLATE_1 [cerr: No error][aaa: Success]
  event Session-Activate match-first [at Wed May 13 10:40:43 2015]
    class type control subscriber PPP_CM do-all [Succeeded]
      10 authenticate aaa list default [cerr: No error][aaa: Success]
      20 activate dynamic-template DEF_SEVICE [cerr: No error][aaa: Success]
Session Accounting:
 Acct-Session-Id:          10000003
 Method-list:              default
 - - -
 - - -
Last COA request received: unavailable
User Profile received from AAA:
 Attribute List: 0x1000f524
1:  service-type    len=  4  value= Framed
2:  inacl           len= 18  value= ACL_VOZ_CONTROL_IN
Services:
  Name         : PTA_TEMPLATE_1
  Service-ID   : 0x4000002
  Type         : Template
  Status       : Applied
-------------------------
  Name         : 2_Mbps_IN
  Service-ID   : 0x400001d
  Type         : Profile
  Status       : Applied
```

RADIUS Based Policing - QoS Shaper Parameterization

Radius Based Policing (RaBaPol) allows customized parameters, instead of the default parameters, to be used to activate BNG subscriber services. BNG supports parameterization of QoS **shape-rate**. The shaper parameters can either be sent to BNG by the RADIUS server during connection establishment, as CISCO VSAs in an Access Accept message, or they can be sent to BNG as part of the CoA messages.

To configure QoS Shaper Parameterization, use the shape average $*var_name* = *value* command in policy-map class configuration mode.

According to RaBaPol, the dynamic template associated with the subscriber contains individual feature configuration. The syntax and semantics of parameterization is feature dependent. For QoS, a dollar sign ($) is added as a prefix to the **shape-rate** variable, and the default value, along with the variables, is configured in the policy-map definition.

If the service that is to be activated is already associated to the subscriber, the incoming variable-list is compared with the exiting one. If the variable-list is the same, then this is a duplicate request and the request gets dropped. Otherwise, the old variable-list is cached and the new variable-list is associated to the subscriber. After the service is successfully activated, the iEdge echoes the VSA that trigged the service-activate, as an acknowledgment back to the AAA server.

If any feature returns an error during its activation, the iEdge component rollbacks all features to their previous states. If the feature or service has a variable-list associated with it, then that variable-list is also rolled back to the previous cached variable-list.

RaBaPol also supports policy merge, where QoS policies from multiple dynamic templates (configured through CLI or downloaded from AAA server) are merged for the subscriber.

High Availability - In the case of process restart, the session is re-established using the variable-list that is already associated with the service.

Sample Configuration and Use Cases for QoS Shaper Parameterization

Sample Configuration for QoS Shaper Parameterization

This is a sample configuration for QoS Shaper Parameterization:

```
dynamic-template type service SERVICE-POLICY-OUT
    service-policy output out-policy merge 10

policy-map out-policy
    class class-default
       shape average $shape-rate= 100000 Kbps
service-policy output-child
policy-map output-child
      class class-default
```

In this example, the service named SERVICE-POLICY-OUT has QoS features enabled. This dynamic template has outgoing QoS policies configured, with a default value of **shape-rate** being 100 Mbps.

Use Cases for QoS Shaper Parameterization

These are some use cases for QoS Shaper Parameterization:

- User initiates a subscriber session with this user profile:

```
user-cpe-xyz1@abc.com         Password="abc"
        Framed-Protocol=PPP,
        Service-Type=Framed-User
        …..
        Cisco-avpair = "subscriber:sa=SERVICE-POLICY-OUT(shape-rate=1203000)”
```

  The AAA server sends to BNG an Access-accept message that contains the service name that is to be activated (SERVICE-POLICY-OUT, in this example), action type (subscriber:sa) , and the variable list, along with its values. Now, the service name maps with the dynamic-template defined on BNG. The

VSA contains QoS **shape-rate** value (For example, shape-rate=1203000) to override the default values locally configured on BNG. In BNG, the policy gets merged with default and customized values. For the variables that were not specified in the AAA message, default values are retained.

Alternatively, the new service activation can be performed using CoA. In this case, the old policy is removed and the new, merged policy gets configured in the hardware.

- User wants to change the QoS shaper value of the subscriber. This can be ideally be done in two ways:
  - Service-modify of same service - This is currently not supported.
  - Service-activate of the new service followed by Service-deactivate of the old service - At first, a new service is activated using the new shaper value sent through the Access-accept message. After that, a CoA message is sent from the AAA server to the BNG, to deactivate the old service.

Verification of QoS Shaper Parameterization Configurations

These show commands can be used to verify the QoS Shaper Parameterization configurations in BNG:

SUMMARY STEPS

1. **show policy-map interface all**
2. **show policy-map applied interface** *interface-type interface-name*
3. **show running-configuration policy-map**
4. **show qos-ea interface** *interface-type interface-name*

DETAILED STEPS

Step 1 **show policy-map interface all**

Displays the QoS shaper rate configured on the subscriber interface by the AAA server, either through an Access-Accept message or through a CoA message. The statistics rate field, transmitted, displays the shaper rate.

Example:

```
RP/0/RSP0/CPU0:router#
show policy-map interface all
node0_1_CPU0: Service Policy not installed
node0_0_CPU0: Service Policy not installed
node0_RSP1_CPU0: node0_RSP0_CPU0:
Bundle-Ether1.1.pppoe62151: policy-parent
Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :                   0/0                    0
    Transmitted         :                   0/0                    0
    Total Dropped       :                   0/0                    0
  Queueing statistics
    Queue ID                             : 458
    High watermark                       : N/A
    Inst-queue-len  (packets)            : 0
    Avg-queue-len                        : N/A
    Taildropped(packets/bytes)           : 0/0
    Queue(conform)      :                   0/0                    0
    Queue(exceed)       :                   0/0                    0
```

```
    RED random drops(packets/bytes)      : 0/0
```

Step 2 **show policy-map applied interface** *interface-type interface-name*

Displays the actual policy-map applied on the subscriber interface.

Example:

```
RP/0/RSP0/CPU0:router#
show policy-map applied interface Bundle-Ether1.1.pppoe62151
Output policy-map applied to Bundle-Ether1.1.pppoe62151:
   policy-map policy-parent
   class class-default
   service-policy policy-child
    shape average $shaperP = 500 mbps
   !
 Child policy-map(s) of policy-map policy-parent:

  policy-map policy-child
   class prec2
    shape average $shaperC1 = 600 kbps
   !
   class prec3
    shape average $shaperC2 = 700 kbps
   !
   class class-default
    shape average $shaperC3 = 200 kbps
   !
   end-policy-map
```

Step 3 **show running-configuration policy-map**

Displays the details of the policy-map configured on BNG.

Example:

```
RP/0/RSP0/CPU0:router#
show running-configuration policy-map
policy-map policy-parent
class class-default
  service-policy policy-child
  shape average $shaperP = 500 mbps
!
end-policy-map
!

policy-map policy-child
class prec2
  shape average $shaperC1 = 600 kbps
!
class prec3
  shape average $shaperC2 = 700 kbps
!
class class-default
  shape average $shaperC3 = 200 kbps
!
end-policy-map
!
```

Step 4 **show qos-ea interface** *interface-type interface-name*

Displays the QoS programmed in the hardware.

Example:

```
RP/0/RSP0/CPU0:router#
show qos-ea interface bundle-Ether 1.4 output member gigabitEthernet 0/1/0/0

Interface: GigabitEthernet0_1_0_0 output policy: vlan_policy_egress
Total number of classes:   1
Total number of UBRL classes:  0
Total number of CAC classes:   0
-------------------------------------------------------
Policy name: vlan_policy_egress
Hierarchical depth 1
Interface type VLAN Subif
Interface rate 1000000 kbps
Port Shaper rate 0 kbps
Interface handle 0x00096060
ul_ifh 0x060000C0, ul_id  0x00000000
uidb index 0x001B
qos_ifh 0x810800000001b
Local port 0, NP 0
Policy map id 0x2014, format 16, uidb index 0x001B
-------------------------------------------------------
 Index 0 Level 0 Class name class-default service_id 0x0 Policy name vlan_policy_egress
 Node flags: LEAF Q_LEAF DEFAULT DEFAULT-ALL
 Stats flags: Queuing enabled
 Node Config:
 Shape: CIR/CBS/PIR/PBS: 0kbps/11250000B/900000kbps/11250000B
 WFQ: BW/Sum of BW/Excess ratio: 0kbps/0kbps/1
 Queue limit 11250000 Guarantee 0
 Node Result: Class-based stats:Stat ID 0x005114F3
 Queue: Q-ID 0x00030062 Stat ID(Commit/Excess/Drop): 0x006E01EA/0x00000000/0x006E01EB
-------------------------------------------------------
```

Supported Scenarios of QoS Shaper Parameterization

These scenarios are supported for QoS Shaper Parameterization:

- Merging of QoS policies through AAA server is supported - A subscriber session does not come up if only one of the policies (applied either through dynamic template control policy or through RADIUS server) has the **merge** keyword enabled. This is irrespective of whether the shaper parameterization is enabled, or not.

  In the case of re-configuring through a CoA message:

  - If only one of the policies has the **merge** keyword enabled, the policy is rejected irrespective of whether the shaper parameterization is enabled, or not.
  - If none of the policies have the **merge** keyword enabled, the policy applied through the RADIUS server replaces the one applied through the control subscriber policy.
  - If both the policies have the **merge** keyword enabled, the policy is accepted irrespective of whether the shaper parameterization is enabled, or not.

- These service policy replacement scenarios are supported:

- A subscriber, with service policy A (applied through dynamic template) having default shaper parameterized values, which is replaced by service policy B having shaper parameterized values.
- A subscriber, with service policy A (applied through dynamic template) having default shaper parameterized values, which is replaced by service policy B with no shaper parameterized values.

Restrictions of QoS Shaper Parameterization

The QoS Shaper Parameterization is subjected to these restrictions:

- Parameterization of only the QoS *shape-rate* feature associated with the subscriber service is supported; other features are not supported. For the *shape-rate* variable, the parameterization of only the value attribute is supported; the parameterization of rate units (such as kbps, mbps, and so on) including excess burst size, is not supported.
- Linecard (LC) subscribers are not supported.
- The service profile downloaded from the RADIUS server is not supported.
- The addition or modification of the *shape-rate* variable field is rejected for any policy-map that is applied on an interface.
- The modification of the default *shape-rate* variable value is rejected for a policy-map that is applied on an interface.
- The variable names in the policy-map definition must be different across the system.
- A service which is defined on BNG, without parameterization enabled, does not accept parameters through CoA.
- Service modification of variable-list is not supported.
- Parameterized shapers are not supported with multi-action CoA.
- The maximum number of different variable-lists supported is 2000. This limit includes the already-active sessions wherein the previous variable-list is also stored.
- Only absolute shaper values is supported in the highest level of the policy. At the child level of the policy, the absolute and the percent-based shaper values can be configured.
- Shared Policy Instance (SPI) is not supported on parameterized shaper policies.
- Scenarios with Parameterized (PQoS) policy-map and CLI policy-map applied through the RADIUS server, are not supported.
- Service Accounting - For a subscriber with service policy A (applied through dynamic template), a service policy-replacement with service policy B, is not supported (irrespective of whether shaper parameterized variables are present in both A or B) for all these scenarios:
    - If service accounting is enabled either in service policy A or in service policy B.
    - If service accounting is enabled in both A and B.
- If more than one service with service accounting is configured for each subscriber session, one of the services must show aggregate traffic of all the services. To have proper per-service accounting, it is recommended to define a parent service in such scenarios, with all the other services defined as children.

QoS Accounting

The QoS overhead accounting feature enables BNG to account for various encapsulation types when applying QoS to packets. The ATM overhead accounting enables the BNG to account for the ATM encapsulation on the subscriber line. It also accounts for the overhead added by cell segmentation. This accounting enables the service provider to prevent overruns on the subscriber line and ensures that the BNG executes QoS features on the actual bandwidth allocated to the subscriber traffic. The ATM overhead encapsulation details are listed in this table.

Table 9: ATM Overhead Encapsulation Details

DSLAM to CPE Encapsulation		ALE Tags (RFC 4679)		
CLI Option	**Overhead (in bytes)**	**Data Link**	**Encapsulation1**	**Encapsulation2**
snap-pppoa	12	AAL5	N/A	PPPoA LLC (1)
mux-pppoa	10	AAL5	N/A	PPPoA Null (2)
snap-1483routed	18	AAL5	Untagged Ethernet	IPoA LLC (3)
mux-1483routed	8	AAL5	Untagged Ethernet	IPoA NULL (4)
snap-rbe	28	AAL5	Untagged Ethernet	Ethernet over AAL5 LLC without FCS (6)
snap-dot1q-rbe	32	AAL5	Single-Tagged Ethernet	Ethernet over AAL5 LLC without FCS (6)
mux-rbe	24	AAL5	Untagged Ethernet	Ethernet over AAL5 Null without FCS (8)
mux-dot1q-rbe	28	AAL5	Single-Tagged Ethernet	Ethernet over AAL5 Null without FCS (8)

To enable QoS overhead accounting, see Configuring QoS Accounting, on page 200.

Configuring QoS Accounting

Perform this task to enable QoS Layer2 overhead accounting.

SUMMARY STEPS

1. **configure**
2. **dynamic-template**
3. **type** [**ppp** | **ip-subscriber** | **service**] *name*
4. **qos-account** [**AAL5** | **user-defined**] [**mux-1483routed** | **mux-dot1q-rbe** | **mux-pppoa** | **mux-rbe** | **snap-1483routed** | **snap-dot1q-rbe** | **snap-ppoa** | **snap-rbe**]

5. **exit**
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dynamic-template** **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template`	Enters dynamic template configuration mode.
Step 3	**type** [**ppp** | **ip-subscriber** | **service**] *name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ppp p1`	Specifies the type of dynamic template that needs to be applied. Three type are: • PPP • IP-subscriber • Service
Step 4	**qos-account** [**AAL5** | **user-defined**] [**mux-1483routed** | **mux-dot1q-rbe** | **mux-pppoa** | **mux-rbe** | **snap-1483routed** | **snap-dot1q-rbe** | **snap-ppoa** | **snap-rbe**] **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# qos-account AAL5 snap-rbe`	Defines the L2 QoS overhead accounting. Various keywords such as mux-1483routed, snap-rbe define different available encapsulations between the DSLAM and CPE. For details about keywords, see Table 9: ATM Overhead Encapsulation Details, on page 200.
Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# exit`	Exits from the current mode.
Step 6	**commit**	

Configuring QoS Accounting: An example

```
configure
dynamic-template type ppp p1
qos account AAL5  mux-1483routed
service-policy input input_1
end
```

Support for Shared Policy Instance

Shared Policy Instance (SPI) allows allocation of a single set of QoS resources among groups of BNG sub-interfaces and bundle sub-interfaces, and shares them across a group of sub-interfaces, multiple Ethernet flow points (EFPs), or bundle interfaces.

Using SPI, a single instance of QoS policy can be shared across multiple sub-interfaces, allowing for aggregate shaping of the sub-interfaces to one rate. All sub-interfaces that share the instance of a QoS policy must belong to the same physical interface. The number of sub-interfaces sharing the QoS policy instance can range from 2 to the maximum number of sub-interfaces on the port.

For bundle interfaces, hardware resources are replicated per bundle member. All sub-interfaces that use a common shared policy instance and are configured on a Link Aggregation Control Protocol (LAG) bundle must be load-balanced to the same member link.

When a policy is configured on a bundle EFP, one instance of the policy is configured on each of the bundle member links. When using SPI across multiple bundle EFPs of the same bundle, one shared instance of the policy is configured on each of the bundle member links. By default, the bundle load balancing algorithm uses hashing to distribute the traffic (that needs to be sent out of the bundle EFPs) among its bundle members. The traffic for single or multiple EFPs can get distributed among multiple bundle members. If multiple EFPs have traffic that needs to be shaped or policed together using SPI, the bundle load balancing has to be configured to select the same bundle member (hash-select) for traffic to all the EFPs that belong the same shared instance of the policy. This ensures that traffic going out on all the EFPs with same shared instance of the policy use the same policer or shaper Instance.

BNG configures a complete hierarchical policy-map that includes parent and child policies. Optionally, the SPI name can be defined and attached to the appropriate dynamic template or downloaded from RADIUS, in this manner:

- Policy configured through a CLI and applied through a dynamic-template
- Policy configured through a CLI and applied through RADIUS

Note The SPI has to be used across subscriber interfaces, but all subscriber interfaces have to be under the same access-interface (that is, the parent interface has to be same).

Restrictions

These restrictions apply to the usage of shared policy instance:

- SPI is not supported for subscribers on non-bundle interfaces.
- SPI is not supported for Parameterized QoS (PQoS). In a PQoS configuration, if there exists a SPI name, then it is ignored.
- Prior to Cisco IOS XR Release 5.2.0, SPI modified through CoA is not supported on subscribers.
- The SPI name must be changed if the policy-map associated with it is changed.
    - Once an SPI policy has been applied on a subscriber, a new policy with same policy-map name and different SPI name is rejected.

- Once an SPI policy has been applied on a subscriber, a new policy with different policy-map name and same SPI name is rejected.

Configuring a Policy with SPI in the Input or Output Direction Using Dynamic Template

Perform this task to configure a policy with shared policy instance in the input and output direction using dynamic template.

SUMMARY STEPS

1. **configure**
2. **policy-map** *policy_map_name*
3. **class** {*class_name* | **class-default** | } [**type qos**]
4. **service-policy** *service_policy_name*
5. **commit**
6. **policy-map** *policy_map_name*
7. **class** {*class_name* | **class-default** | } [**type qos**]
8. **police rate** *value*
9. **commit**
10. **dynamic-template type ipsubscriber** *dynamic_template_name*
11. **service-policy** {**input** | **output**}*policy_map_name* [**shared-policy-instance** *instance_name*]
12. **service-policy** {**input** | **output**}*policy_map_name* [**shared-policy-instance** *instance_name*]
13. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map** *policy_map_name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map policy1`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy, and enters the policy-map configuration submode.
Step 3	**class** {*class_name* | **class-default** | } [**type qos**] **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	Specifies the name of the class whose policy you want to create or change and enters the policy map class configuration submode. This example configures a traffic policy for the default class of the traffic policy policy1. The default class is named class-default.
Step 4	**service-policy** *service_policy_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy policy1_child`	Attaches a policy map to an input or output interface.

	Command or Action	Purpose
Step 5	**commit**	
Step 6	**policy-map** *policy_map_name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map policy1_child`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy, and enters the policy-map configuration submode.
Step 7	**class** {*class_name* | **class-default** | } [**type qos**] **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	Specifies the name of the class whose policy you want to create or change and enters the policy map class configuration submode. This example configures a traffic policy for the default class of the traffic policy policy1. The default class is named class-default.
Step 8	**police rate** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# police rate 1024`	Configures traffic policing and enters policy map police configuration mode. The value represents the committed information rate and ranges from 1 to 4294967295.
Step 9	**commit**	
Step 10	**dynamic-template type ipsubscriber** *dynamic_template_name* **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template type ppp PTA_TEMPLATE_1`	Creates a dynamic template of type ipsubscriber.
Step 11	**service-policy** {**input** | **output**}*policy_map_name* [**shared-policy-instance** *instance_name*] **Example:** `RP/0/RSP0/CPU0:router(config)# service-policy input policy1 shared-policy-instance spi_1`	Attaches a policy map to an input or output interface to be used as the service policy for that interface. In this example, the traffic policy evaluates all traffic entering into that interface.
Step 12	**service-policy** {**input** | **output**}*policy_map_name* [**shared-policy-instance** *instance_name*] **Example:** `RP/0/RSP0/CPU0:router(config)# service-policy output policy1 shared-policy-instance spi_2`	Attaches a policy map to an input or output interface to be used as the service policy for that interface. In this example, the traffic policy evaluates all traffic leaving that interface.
Step 13	**commit**	

Configuring a Policy with SPI in the Input or Output Direction Using Dynamic Template: Example

```
configure
policy-map policy1
```

```
class class-default
service-policy policy1_child
!!

policy-map policy1_child
class class-default
police rate 1024 kbps
!!

dynamic-template
type ppp PTA_TEMPLATE_1
service-policy input policy1 shared-policy-instance spi_1
service-policy output policy1 shared-policy-instance spi_2
commit
```

Configuring a Policy with SPI in the Input or Output Direction Using RADIUS

Perform this task to configure a policy with shared policy instance in the input or output direction using RADIUS.

SUMMARY STEPS

1. **configure**
2. **policy-map** *policy_map_name*
3. **class** {*class_name* | **class-default**} [**type qos**]
4. **service-policy** *service_policy_name*
5. **commit**
6. **policy-map** *policy_map_name*
7. **class** {*class_name* | **class-default**} [**type qos**]
8. **police rate** *value*
9. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map** *policy_map_name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map policy1`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy, and enters the policy-map configuration submode.
Step 3	**class** {*class_name* | **class-default**} [**type qos**] **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	Specifies the name of the class whose policy you want to create or change and enters the policy map class configuration submode. This example configures a traffic policy for the default class of the traffic policy policy1. The default class is named class-default.
Step 4	**service-policy** *service_policy_name* **Example:**	Attaches a policy map to an input or output interface.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy policy1_child`	
Step 5	**commit**	
Step 6	**policy-map** *policy_map_name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map policy1_child`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy, and enters the policy-map configuration submode.
Step 7	**class** {*class_name* | **class-default**} [**type qos**] **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	Specifies the name of the class whose policy you want to create or change and enters the policy map class configuration submode. This example configures a traffic policy for the default class of the traffic policy policy1. The default class is named class-default.
Step 8	**police rate** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# police rate 1024`	Configures traffic policing and enters policy map police configuration mode. The value represents the committed information rate and ranges from 1 to 4294967295.
Step 9	**commit**	

Configuring a Policy with SPI in the Input or Output Direction Using RADIUS: Example

```
configure
policy-map policy1
class class-default
service-policy policy1_child
!!

policy-map policy1_child
class class-default
police rate 1024 kbps
commit
!!

//In the USER file in RADIUS
RoadRunner_P1@Chasing1 Cleartext-Password := "LooneyTunes_P1"
cisco-avpair += "sub-qos-policy-in=policy1 shared-policy-instance spi_1",
cisco-avpair += "sub-qos-policy-out=policy1 shared-policy-instance spi_2",
Framed-Protocol += PPP,
Service-Type += Framed-User,
Fall-Through = no
```

What to do next

Run these steps in the USER file in RADIUS:

```
RoadRunner_P1@Chasing1 Cleartext-Password := "LooneyTunes_P1"
```

```
cisco-avpair += "sub-qos-policy-in=policy1 shared-policy-instance spi_1",
cisco-avpair += "sub-qos-policy-out=policy1 shared-policy-instance spi_2",
Framed-Protocol += PPP,
Service-Type += Framed-User,
Fall-Through = no
```

Merging QoS Policy-maps

Multiple QoS policies, applied through multiple dynamic templates, can be merged and implemented on a single subscriber. The order in which the policies are merged is important, and is determined by the value of the sequence number configured in the dynamic template. A policy is deployed using a policy-map. A new optional **merge** keyword is provided with the **service-policy** command under dynamic template sub mode to allow the merging of policy-maps applied through multiple dynamic templates.

When more than two policy-maps are to be merged, two policy-maps are first merged together based on their sequence number (using the rules listed below) to create a merged policy-map. Similarly, a third policy-map is merged with the first merged policy-map. This continues till all policy-maps that are to be merged are merged together. The sequence numbers of the policy-maps are significant only while merging the policy-maps and are not related to the priority of the classes.

For example, let's say that policy-maps *p1, p2, p3, p4* each with a specific sequence number, are to be merged in that order; *p1* and *p2* are merged first based on their sequence numbers (see the rules listed below). Next, *p3* is merged with the <p1-p2> merged policy-map. Finally, *p4* is merged with the <p1-p2-p3> merged policy-map, giving the final merged policy-map.

The rules for merging two policy-maps are:

- The policy-map with the lowest sequence number takes precedence over the other.

  For example, if policy-maps *p1* with a sequence number 100 and *p2* with a sequence number 20 are to be merged, then *p2* overrides *p1*, as it has the lowest sequence number. Similarly, if policy-map *p3* with a sequence number 10 has to be merged with the policy-map *p2* having sequence number 20, then the policy-map *p3* takes precedence over the other, as it has the lowest sequence number.

- If the same class (except for the default class) is configured under both the policies, the instance of that class (including all actions configured under it) in the second policy is ignored.

- If the default class under the first policy contains any actions other than any child policy actions, then that default class is added to the end of the merged policy. If it contains any child policy actions, then the default class from the second policy is added at the end of the merged policy.

- If a child policy is configured under the default class of both policies, the two child policies are merged using the rules above. The merged child policy is then applied as the child policy under the default class of the merged parent policy.

- If a child policy is configured under the default class of either the first or second policy (but not both), then it is applied (as it is) as the child policy under the default class of the merged policy. Child policies under classes other than the default class are never merged together.

Note If the sequence numbers of two policies to be merged are configured to be the same, the order in which they are merged with respect to each other is random, and may change after the process restarts. Such configurations must be avoided.

Enabling Policy-maps Merge

Perform this task to enable merging of multiple QoS policy-maps applied through multiple dynamic templates.

SUMMARY STEPS

1. **configure**
2. **dynamic-template**
3. **type service** *dynamic-template-name*
4. **service-policy** {**input** | **output** | **type**} *service-policy_name* [**acct-stats**] [**merge** *seq_num*]
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dynamic-template** **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template`	Enters the dynamic-template configuration mode.
Step 3	**type service** *dynamic-template-name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type service s1`	Creates a dynamic-template with a user-defined name for a service.

Step 4	**service-policy** {**input** | **output** | **type**} *service-policy_name* [**acct-stats**] [**merge** *seq_num*]
	Example:
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy input QoS1 merge 10`
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# service-policy output QoS2 merge 20`
		Associates a service-policy to the dynamic template, and enables merging of multiple QoS policies.

Step 5	**commit**	

Enabling Policy-maps Merge: Examples

```
dynamic-template type service default-service
   service-policy input default-policy-in merge 100
   service-policy output default-policy-out merge 100
!
dynamic-template type service voip-service
   service-policy input voip-policy-in merge 20
   service-policy output voip-policy-out merge 30
!
dynamic-template type service vod-service
   service-policy input vod-policy-in merge 30
```

```
   service-policy output vod-policy-out merge 50
!
dynamic-template type service turbo-button-service
   service-policy input turbo-button-policy-in merge 10
   service-policy output turbo-button-policy-out merge 10
!
end

\\the following configuration explains the merging behavior of egress qos policies
policy-map type qos default-policy-out
  class class-default
      shape average 2 mbps
      bandwidth 512 kbps
    service-policy default-policy-child-out
  !
  end-policy-map

policy-map type qos default-policy-child-out
   class critical-data
     bandwidth percent 90
     set cos 3
     queue-limit 500 ms
   !
   class best-effort-data
     shape average percent 50
     random-detect 100 ms 200 ms
      set cos 5
    !
    class class-default
      shape average percent 20
      set cos 7
    !
    end-policy-map

policy-map type qos voip-policy-out
   class class-default
     service-policy voip-policy-child-out
   !
  end-policy-map

 policy-map type qos voip-policy-child-out
   class voip-control
     priority level 1
     set cos 2
   !
   class voip-data
     priority level 2
     set cos 2
     random-detect 100 ms 200 ms
  !
   class class-default
   !
   end-policy-map

policy-map type qos vod-policy-out
  class class-default
    service-policy vod-policy-child-out
  !
  end-policy-map


policy-map type qos vod-policy-child-out
  class vod-control
     priority level 1
```

```
      set cos 1
    !
    class vod-data
     priority level 2
     queue-limit 100 ms
    !
    class class-default
    !
     end-policy-map

policy-map type qos turbo-button-policy-out
  class class-default
     shape average 10 mbps
     bandwidth 2 mpbs
   !
   end-policy-map


\\after the default and voip services are enabled on a subscriber session

policy-map type qos  <merged-policy-1>   !! Name is generated internally. This is just an
example.
  class class-default
      shape average 2 mbps
      bandwidth 512 kbps
      service-policy <merged-child-policy-1>
    !
   end-policy-map

policy-map type qos <merged-child-policy-1>
   class voip-control
     priority level 1
     set cos 2
   !
   class voip-data
     priority level 2
     set cos 2
     random-detect 100 ms 200 ms
  !
   class critical-data
     bandwidth percent 90
     set cos 3
     queue-limit 500 ms
   !
   class best-effort-data
     shape average percent 50
     random-detect 100 ms 200 ms
      set cos 5
    !
    class class-default
      shape average percent 20
      set cos 7
    !
    end-policy-map

\\after the turbo-button service is enabled

policy-map type qos <merged-policy-2>
   class class-default
     shape average 10 mbps
     bandwidth 2 mpbs
    service-policy <merged-child-policy-1> !! <merged-child-policy-1> is the same as before
 since the
                                            !! the turbo-button-policy-out does not have
```

```
any child policy
                                            !! to be merged.
   !

\\after the vod service is enabled

policy-map type qos  <merged-policy-3>
  class class-default
      shape average 10 mbps
      bandwidth 2 mbps
      service-policy <merged-child-policy-2>
    !
   end-policy-map

policy-map type qos <merged-child-policy-1>
   class voip-control
     priority level 1
     set cos 2
   !
   class voip-data
     priority level 2
     set cos 2
     random-detect 100 ms 200 ms
  !
  class vod-control
     priority level 1
     set cos 1
  !
  class vod-data
   priority level 2
   queue-limit 100 ms
  !
   class critical-data
     bandwidth percent 90
     set cos 3
     queue-limit 500 ms
   !
   class best-effort-data
     shape average percent 50
     random-detect 100 ms 200 ms
      set cos 5
    !
    class class-default
      shape average percent 20
      set cos 7
    !
    end-policy-map
```

QoS Features Supported on BNG

BNG supports these QoS features:

Policing and Queuing Support

BNG provides ingress and egress traffic policers. BNG also supports pre-existing traffic policing mechanisms per subscriber session. 1R2C and 2R3C policers with marking actions is supported at parent-level in subscriber policies. Only absolute police rates are supported at the parent-level of subscriber policies. 1R2C and 2R3C policers with marking actions are supported at the child-level in subscriber policies. Both absolute and percentage based police rates are supported at child-level of subscriber policies.

BNG supports traffic shaping at the physical port level, at the subscriber session level, at the class level, and at the VLAN level only in egress direction. The system supports all pre-existing queuing actions for subscriber sessions. The configuration of minimum-bandwidth at the parent-level in subscriber policies is blocked. If subscriber policies do not have a queuing action, the traffic on those subscribers is still subjected to S-VLAN shaping and the traffic goes out through S-VLAN policy queues if those are present; if not, the traffic goes through the interface default-queue. The shaping or bandwidth-remaining queuing action is mandatory in flat S-VLAN policies. Only absolute shape rates is supported in S-VLAN flat policies and the parent-level of subscriber policies. However, only shaping and bandwidth-remaining queuing actions are supported in the parent-level of subscriber policies and all queuing actions are supported in the child-level of subscriber policies.

These additional queuing features are supported in egress policies applied on subscribers:

- A policy can have 1 P1, 1 P2, 1 P3 and 5 normal priority queues.
- A policy can have 1 P1, 2 P2 and 5 normal priority queues. P1 and P3 queues can be shared by multiple classes whereas P2 queues are never shared.

Default Marking

BNG supports all pre-existing classification and marking options supported for L3 interfaces for use with subscriber sessions. BNG also supports L3 marking to L2 marking mapping. BNG also supports ToS to CoS mapping at LAC for downstream PPPoE frames and provides mechanisms to mark 802.1p and IP TOS fields. The system allows flexible IP TOS marking for L2TP packets based on ingress subscriber qos policy. Marking is supported at the parent-level in subscriber policies and at the child-level in subscriber policies.

QoS Policy Modification

BNG supports in-service QoS policy-modification. Modification of subscriber-policy (through Radius), S-VLAN policy (through CLI) and port sub-rate policy (through CLI) are also supported.

L2 Encapsulation

For PPPoE subscribers, the L2 encapsulation size used in QoS rate calculations must be adjustable based on the last mile encapsulation (DSLAM to subscriber home) signaled in the PPPoE tags.

Classification

The BNG supports all pre-existing classification and marking options supported for L3 interfaces for use with subscriber sessions. BNG also supports ingress classification based on 802.1P values for single and double tagged COS, classification based on DSCP in either direction, classification based on L3/L4 ACLs in either direction, and classification of L2TPv2 traffic based on the outer DSCP marking.

The classification of an incoming L2TP packet on the ingress core side interface is always based on the outer IP fields even if the packet arrives with an MPLS tag stack.

Policy Inheritance

This table is relevant for egress direction only, as in ingress direction sub-rate policy and S-VLAN policy is not supported:

Port	S-VLAN	Subscriber
Sub-rate policy	No policy is configured. Inheritance limited to traffic getting shaped by port sub-rate policy. This is done irrespective of whether a policy is configured on the S-VLAN, or not.	Subscriber policy , if present, is executed first; then, traffic is subjected to port-shaper.
Sub-rate policy	Policy is configured. Inheritance limited to traffic that gets shaped by port sub-rate policy. This is done irrespective of whether a policy is configured on the S-VLAN or not.	Subscriber policy is executed first, if present, and then, S-VLAN policy is executed. Finally traffic is subjected to port-shaper.
HQoS or policy with more than class-default	Policy configuration is blocked and port policy is inherited.	Policy configuration is blocked and port policy inherited through the S-VLAN.
No policy configured	Policy is configured.	Subscriber policy is executed first, if present, and then S-VLAN policy is executed.

Subscriber with No QoS

When QoS is not configured on a subscriber, the parent S-VLAN, or on the port, subscriber traffic goes out using the default-queue of its parent's physical port.

- The subscriber is subjected to the S-VLAN policy and goes out using S-VLAN policy queues, if those are present. If the S-VLAN policy does not have its own queues, then all the S-VLAN traffic, including the subscriber's, goes out through the default queue of the physical interface.
- The subscriber is subject to a port policy, but no S-VLAN policy. Similar to the S-VLAN case, the subscriber traffic is subject to it and uses its queues.
- If a non-port-shaper policy is applied on the port, the application of policy on S-VLAN and subscriber is blocked. In such a scenario, subscriber traffic is subjected to the policy applied on the port.

Control Packet Handling

BNG provides priority treatment in handling PPP Link Control Protocol (LCP) packets. The control packets are handled in high priority without the need of user configuration, and these packets are not subjected to QoS policies that are applied on both ingress and egress of the interface. In the case of LAC upstream direction, if user wants a trusted COS value, then a PPP command is provided to impose the core-side header based on the set trusted-COS. Thus, this ensures the priority treatment of these control packets in the network.

S-VLAN Shaping and Statistics

In the egress direction, the BNG supports the ability to have policies at three different levels: the subscriber interface level, the stacked virtual local area network (S-VLAN), and at the port level. The egress S-VLAN and port-level policies are applied through CLI directly at the interface level. For applying a QoS policy on S-VLAN, see Configuring Policy on S-VLAN, on page 216

The subscriber policy can only be applied through a dynamic template or via RADIUS. The egress subscriber policy can be a two-level policy. The S-VLAN and port-level policies can only be flat policies, with only the class default, with the only action being a shaped rate. Essentially it provides a means to constrain the S-VLAN or port to a maximum rate via shaping.

In the ingress direction, the traffic is only subject to the subscriber input policy where the subscriber policies are applied through RADIUS or dynamic-template.

The traffic through the S-VLAN includes traffic to many subscribers that may have already been shaped by the subscriber policies. Providing statistics on that S-VLAN shaper is important in order to monitor whether it is reaching the maximum capacity. Unlike the subscriber QoS policies, the HW does not have the ability to directly track the usage or transmitted packets/bytes through this S-VLAN shaper. So unlike other statistics, the BNG provides the S-VLAN QoS policy-related statistics by aggregating the statistics of the underlying subscriber policies. The statistics are displayed via show commands (and MIBs as appropriate) consistent with all other interface types.

S-VLAN supports these conditions:

- Modification of QoS rates.
- Modification of S-VLAN policy to change number of levels in the policy is rejected.
- Modification of two-level S-VLAN policy to add or remove child-level classes is rejected.
- Modification of classification criteria in child-level classes, in two-level policy, is rejected.
- Addition or removal of actions, in both two-level and flat policy, is rejected.

QoS Attachment Points

This table lists the QoS attachment points, and modes for definition and application.

QoS Attachment Point	Definition	Application	Type of Policy
Port (sub-rate policy)	CLI/XML	CLI/XML	Flat – class-default only
S-VLAN	CLI/XML	CLI/XML	Flat – class-default only. 2 level, with parent class-default only and child any classification.
Subscriber	CLI/XML	Dynamic-Template	2 level, with parent class-default only and child any classification.
Subscriber	CLI/XML	RADIUS	2 level, with parent class-default only and child any classification.
Subscriber	RADIUS (parameterized QoS)	RADIUS	2 level, with parent class-default only and child any classification.

Un-supported configurations will not be blocked. In S-VLAN policies and subscriber policies, any configuration other than the ones listed in these tables will be blocked:

Table 10: Supported Configuration in Ingress Direction

	Classification	Action	Rates
Subscriber Parent Level Policy	Class-default only	police, marking	Absolute only
Subscriber Child Level Policy	Any, with baseline restrictions	police, marking	Absolute and percent

Table 11: Supported Configurations in Egress Direction

	Classification	Action	Rates
S-VLAN Flat Policy	Class-default only	Any, with mandatory shape action	Absolute only
S-VLAN Parent Level Policy	Class-default only	Any, with mandatory shape action	Absolute only
S-VLAN Child Level Policy	Any, with baseline restrictions	Any	Absolute and percent
Subscriber Parent Level Policy	Class-default only	shape, bandwidth remaining, police, marking	Absolute only
Subscriber Child Level Policy	Any, with baseline restrictions	Any	Absolute and percent

VLAN Policy on Access Interface

BNG supports ingress and egress VLAN policies on an access-interface. Unlike as in the case of S-VLAN (subscriber-parent) policy, the access-interface VLAN policy is not inherited by the session policy. The VLAN policy does not provide reference bandwidth to session policies. The VLAN policy statistics does not include session policy statistics. Only the access-interface traffic is subjected to the VLAN policy.

For details, see Configuring VLAN Policy on an Access Interface, on page 217.

This table summarizes the support for VLAN and S-VLAN policies in ingress and egress directions:

Policy Direction	V-LAN policy (without subscriber-parent keyword)	S-VLAN policy(with subscriber-parent keyword)
Ingress	Supported	Not supported
Egress	Supported	Supported

Restrictions

These restrictions apply to the VLAN policy on the access-interface, when used without the **subscriber-parent** keyword:

- The VLAN policy needs to be attached to the access-interfaces, before bringing up the sessions with QoS policies.
- The restrictions specified for the in-place modification of S-VLAN policy, are applicable to VLAN policy as well. For instance, the in-place modification for the VLAN policy supports only rate-changes. This restriction also applies in adding a policer or shaper and in changing the policy-map to include more classes.

Configuring Policy on S-VLAN

Perform this task to apply a QoS policy on a S-VLAN.

Note

- S-VLAN policy has to be provisioned before any policies are installed on subscribers.
- Application of S-VLAN policy is rejected, if policies are already installed on subscribers.
- Removal of S-VLAN policy is rejected, if subscriber policies are present under that S-VLAN.

SUMMARY STEPS

1. **configure**
2. **interface** *type*
3. **service-policy output** *name* **subscriber-parent**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type* **Example:** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether1.1`	Configures the subscribers on the Bundle-Ether access interface.
Step 3	**service-policy output** *name* **subscriber-parent** **Example:** `RP/0/RSP0/CPU0:router(config-if)# service-policy output svlan subscriber-parent`	Configures the s-vlan policy with the subscriber-parent keyword.
Step 4	**commit**	

Configuring Policy on S-VLAN: An example

```
configure
interface Bundle-Ether1.1
service-policy output svlan_pmap subscriber-parent
end
!
```

Configuring VLAN Policy on an Access Interface

Perform this task to apply an ingress and egress QoS VLAN policy on an access interface.

SUMMARY STEPS

1. **configure**
2. **interface** *type*
3. **service-policy input** *service-policy-name*
4. **service-policy output** *service-policy-name*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type* **Example:** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether18.203`	Configures subscribers on the Bundle-Ether access interface.
Step 3	**service-policy input** *service-policy-name* **Example:** `RP/0/RSP0/CPU0:router(config-subif)# service-policy input mark`	Configures the ingress VLAN QoS policy on the access-interface.
Step 4	**service-policy output** *service-policy-name* **Example:** `RP/0/RSP0/CPU0:router(config-subif)# service-policy output metering`	Configures the egress VLAN QoS policy on the access-interface.
Step 5	**commit**	

Configuring Ingress and Egress VLAN Policies on an Access Interface: Example

```
//Attaching Ingress and Egress VLAN Policies on an Access Interface

configure
interface Bundle-Ether1.1
service-policy input INGRESS_MARKING_POLICING_POLICY
service-policy output VLAN_POLICY
end
!


//Attaching Ingress VLAN Policy and Egress S-VLAN Policies on an Access Interface

configure
interface Bundle-Ether1.2
service-policy input INGRESS_MARKING_POLICING_POLICY
service-policy output S_VLAN_POLICY subscriber-parent
end
!
```

Additional References

These sections provide references related to implementing QoS.

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 7

Configuring Subscriber Features

Subscriber features that are configured on BNG 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, providing multicast services to the subscriber, and so on.

Table 12: Feature History for Configuring Subscriber Features

Release	Modification
Release 6.0.1	Added activating IPv6 router advertisement on an IPv4 subscriber interface enhancements
Release 6.0.1	Added Linking to Subscriber Traffic in a Shared Policy Instance Group feature
Release 6.2.1	These new features were introduced: • IGMP QoS Correlation for IPoE Subscribers • SNMP Lawful Intercept Using Circuit-Id • Controlling Subscriber Plans Using Protocol Options

The subscriber features covered in this chapter are:

• Additional References, on page 278

Excessive Punt Flow Trap

The Excessive Punt Flow Trap feature attempts to identify and mitigate control packet traffic from remote devices that send more than their allocated share of control packet traffic. A remote device can be a subscriber device, a device on a VLAN interface, or a device identified by its source MAC address.

When remote devices send control packet traffic to the router, the control packets are punted and policed by a local packet transport service (LPTS) queue to protect the router's CPU. If one device sends an excessive rate of control packet traffic, the policer queue fills up, causing many packets to be dropped. If the rate from one "bad actor" device greatly exceeds that of other devices, most of the other devices do not get any of their control packets through to the router. The Excessive Punt Flow Trap feature addresses this situation.

Note Even when the Excessive Punt Flow Trap feature is not enabled, the "bad actors" can affect services for only other devices; they cannot bring down the router.

The Excessive Punt Flow Trap feature is supported on both subscriber interfaces, and non-subscriber interfaces such as L2 and L3 VLAN sub-interfaces and bundle virtual interfaces (BVIs). If the source that floods the punt queue with packets is a device with an interface handle, then all punts from that bad actor interface are penalty policed. The default penalty rate, for each protocol, is 10 protocols per second (pps). Otherwise, if the source is a device that does not have an interface handle, then all packets from this bad actor are dropped.

Note In the 4.2.x releases, the Excessive Punt Flow Trap feature was called as "Subscriber Control Plane Policing (CoPP)" that only operated on subscriber interfaces.

Functioning of Excessive Punt Flow Trap Feature

The Excessive Punt Flow Trap feature monitors control packet traffic arriving from physical interfaces, sub-interfaces, BVI, and subscriber interfaces. It divides interfaces into two categories:

- "Parent" interfaces, which can have other interfaces under them.
- "Non-parent" interfaces, which have no interfaces under them.

A physical interface is always a parent interface because it has VLAN sub-interfaces. An L3 VLAN sub-interface can either be a parent or a non-parent interface. If the VLAN sub-interface is enabled for subscribers, then it is a parent interface, otherwise it is a non-parent interface. A subscriber interface (IPoE or PPPoE) is always a non-parent interface.

When a flow is trapped, the Excessive Punt Flow Trap feature tries to identify the source of the flow. The first thing it determines is from which interface the flow came. If this interface is not a "parent" interface, then the feature assumes that it is the end-point source of the flow and penalty policing is applied. The software applies a penalty-policer in the case of a BVI interface also. If the trapped interface is a "parent" interface, then instead of penalizing the entire interface (which would penalize all the interfaces under it), this feature takes the source MAC address of the bad flow and drops all packets from the MAC address under the parent. Due to platform limitation, the penalty policer cannot be applied on a MAC address; therefore all packets are dropped.

For more information about enabling the Excessive Punt Flow Trap feature, see Enabling Excessive Punt Flow Trap Processing, on page 224.

Note The Excessive Punt Flow Trap feature monitors all punt traffic. There is no way to remove a particular interface from the initial monitoring, nor can an interface be prevented from being flagged as bad if it is the source of excessive flows.

Bad actors are policed for each protocol. The protocols that are supported by the Excessive Punt Flow Trap feature are Broadcast, Multicast, ARP, DHCP, PPP, PPPoE, ICMP, IGMP, L2TP and IP (covers many types of L3 based punts, both IPv4 and IPv6). Each protocol has a static punt rate and a penalty rate. For example, the sum total of all ICMP punts from remote devices is policed at 1500 packets per second (pps) to the router's CPU. If one remote device sends an excessive rate of ICMP traffic and is trapped, then ICMP traffic from that bad actor is policed at 10 pps. The remaining (non-bad) remote devices continue to use the static 1500 pps queue for ICMP.

Note The excessive rate required to cause an interface to get trapped has nothing to do with the static punt rate (e.g. 1500 pps for ICMP). The excessive rate is a rate that is significantly higher than the current average rate of other control packets being punted. The excessive rate is not a fixed rate, and is dependent on the current overall punt packet activity.

Once a bad actor is trapped, it is penalty policed on all its punted protocols (ARP, DHCP, PPP, etc.), irrespective of the protocol that caused it to be identified as a bad actor. A penalty rate of 10 pps is sufficient to allow the other protocols to function normally. However, if the bad actor is trapped by source MAC address, then all its packets are dropped.

When an interface is trapped, it is placed in a "penalty box" for a period of time (a default of 15 minutes). At the end of the penalty timeout, it is removed from penalty policing (or dropping). If there is still an excessive rate of control packet traffic coming from the remote device, then the interface is trapped again.

Restrictions

These restrictions apply to implementing Excessive Punt Flow Trap feature:

- The A9K-8x100G-LB-SE and A9K-8x100G-LB-TR line cards do not support BNG subscriber interfaces.
- This feature does not support interfaces on SIP-700 line cards and ASR 9000 Ethernet Line Card.
- This feature is non-deterministic. In some cases, the Excessive Punt Flow Trap feature can give a false positive, i.e. it could trap an interface that is sending legitimate punt traffic.
- The Excessive Punt Flow Trap feature traps flows based on the relative rate of different flows; thus, the behavior depends on the ambient punt rates. A flow that is significantly higher than other flows could be trapped as a bad actor. Thus the feature is less sensitive when there are many flows, and more sensitive when there are fewer flows present.
- Sometimes control packet traffic can occur in bursts. The Excessive Punt Flow Trap has safeguards against triggering on short bursts, but longer bursts could trigger a false positive trap.

MAC-based EPFT on Non-subscriber Interface

This feature supports dropping of the excessive punt packets from a bad actor flow, based on the source MAC address. Before this release, EPFT on non-subscriber interfaces was only performed based on the *ifhandle* (interface handle) of the VLAN sub-interface, wherein all the ingress punt packets on the VLAN sub-interface are penalty policed, irrespective of their source MAC addresses.

In an aggregation scenario, packets may come from multiple source MAC addresses to a VLAN sub-interface. If one particular source MAC sends excessive punt packets, it drains the punt queue; punt packets of other source MAC addresses on that non-subscriber interface may get dropped. MAC-based EPFT on the non-subscriber interface feature performs EPFT (that is, it drops the packets) based on a source MAC address, if the flow is a bad actor flow sending excessive punt packets.

To enable MAC-based EPFT on non-subscriber interface, you must use this command in global configuration mode:

lpts punt excessive-flow-trap non-subscriber-interfaces [**mac**]

Note If the **mac** option is not configured, the default behavior is to perform EPFT, based on the *ifhandle* of the non-subscriber interface.

Tunable Sampler Parameters for Control Plane Policing

This feature allows configuring various EPFT sampler parameters to fine-tune the Elephant Trap algorithm, to achieve the best behavior for realistic traffic streams, and to reduce situations like false positives to a great extent. Before this release, these parameter values were fixed and read from a configuration file.

The commands available for this feature are privileged (Cisco-support) commands.

This table lists configurable EPFT sampler parameters:

EPFT Sampler Parameter	Description
Elephant Trap size	The maximum number of flows that is concurrently stored in Elephant Trap. The range is from 1 to 128; default is 64. The value must be a power of 2, that is 1, 2, 4,8, 16, 32, 64 and 128 are the valid values.
Sampling probability	Sampling probability of Elephant Trap; that is, the probability value to sample any particular packet and feed it into the trap. This is a floating point number ranging from 0 to 1 enclosed in double quotes (""). By default, the value is "0.01", which means that 1 out of 100 packets is randomly picked for sampling.
Report threshold	Threshold at which a flow is reported as a bad actor. The range is from 1 to 65535; default is 5.
Eviction threshold	Threshold below which a flow can be evicted from the Elephant Trap. The range is from 1 to 65535; default is 2.

EPFT Sampler Parameter	Description
Eviction search limit	Maximum number of entries to check before cancelling an eviction search. The range is from 1 to 128; default is 64. Eviction search limit must not be more than the Elephant Trap size.
Maximum flow gap	The maximum time, in milliseconds, that the Elephant Trap allows between successive samples while incrementing the hit counter. The range is from 1 to 60000; default is 800.

False Positive Suppression

Due to the probabilistic nature of the Elephant Trap algorithm, there is possibility of good flows being trapped as bad flows. This probability is more in scenarios where the number of flows is less. Such false positives can be suppressed using these features:

- **Support of tunable sampler parameters for control plane policing**

  For details, see Tunable Sampler Parameters for Control Plane Policing, on page 222.

- **False positive suppression through dampening**

  This feature allows trapping only repeated bad actor flows. The Flowtrap process maintains a trap similar to the Elephant Trap that stores information about each flow for which the bad actor notification is received by the sampler process. The bad actor notifications for penalty policing the flow, or dropping the packets from the flow, is carried out only if the notification is received twice within a specified time (a configurable time in seconds). Although it extends the duration before which a true bad actor is throttled, it also reduces false positives.

  By default, the dampening feature is disabled. To enable this feature, you must use this command in global configuration mode:

  lpts punt excessive-flow-trap dampening [*time*]

  The range of *time* (in milliseconds) is from 1 to 60000. If the *time* option is not used after the **dampening** keyword, a default time value of 30 is used.

EPFT Support for Packet-Triggered Sessions

Before Cisco IOS XR Software Release 5.3.0, punt packets on a packet-triggered subscriber-interface and on a packet-triggered access-interface were policed as per the LPTS rates. The policing rate earlier was high (2000 packets per second) and system wide. With EPFT support for packet triggered sessions, punt packets on packet-triggered interfaces (subscriber and access) go through EPFT node. If identified as bad actor flows, they are penalty-policed according to the EPFT penalty rates (only 20 to 200 packets per second). This is the default behavior from Cisco IOS XR Software Release 5.3.0 and later.

This feature is enabled by default (users need not explicitly configure any command to enable this feature). However, you can use these commands to set the **penalty-rate** and **penalty-timeout** for punt packets of **unclassified-source** type:

lpts punt excessive-flow-trap penalty-rate unclassified *rate*

The range of rate (in pps - packets per second) is from 2 to 100, the default is 10.

lpts punt excessive-flow-trap penalty-timeout unclassified *timeout*

The range of timeout (in minutes) is from 1 to 1000, the default is 15.

Interface-based Flow

For the Elephant Trap sampler, the MAC address is one of the key fields used to uniquely identify a flow. Certain cases of DoS attacks have dynamically changing source MAC addresses. An individual flow does not cross the threshold in such cases, and hence the EPFT does not trap the flow. With the interface-based flow feature, Elephant Trap does not consider MAC addresses as a key for uniquely identifying a flow. Hence, all packets received on a non-subscriber interface (irrespective of the source MAC address) are considered to be a part of a single flow. When excessive punts are received on the interface, EPFT does *ifhandle*-based trap, thereby penalty policing the punt traffic on that particular interface.

To enable interface-based flow, you must use this command in global configuration mode:

lpts punt excessive-flow-trap interface-based-flow

Note You cannot enable this command if EPFT is turned on for the subscriber-interfaces and non-subscriber-interfaces MAC, or vice versa. This is because interface-based flow feature is mutually exclusive with MAC-based EPFT on non-subscriber interface feature.

Enabling Excessive Punt Flow Trap Processing

Perform this task to enable the Excessive Punt Flow Trap feature for both subscriber and non-subscriber interfaces. The task also enables you to set the penalty policing rate and penalty timeout for a protocol.

SUMMARY STEPS

1. **configure**
2. **lpts punt excessive-flow-trap subscriber-interfaces**
3. **lpts punt excessive-flow-trap non-subscriber-interfaces**
4. **lpts punt excessive-flow-trap penalty-rate** *protocol penalty_policer_rate*
5. **lpts punt excessive-flow-trap penalty-timeout** *protocol time*
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**lpts punt excessive-flow-trap subscriber-interfaces** **Example:** `RP/0/RSP0/CPU0:router(config)# lpts punt excessive-flow-trap subscriber-interfaces`	Enables the Excessive Punt Flow Trap feature on subscriber interfaces.
Step 3	**lpts punt excessive-flow-trap non-subscriber-interfaces**	Enables the Excessive Punt Flow Trap feature on non-subscriber interfaces.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router(config)# lpts punt excessive-flow-trap non-subscriber-interfaces`	**Note** If both Step 2 and Step 3 configurations are applied, the Excessive Punt Flow Trap feature is enabled for all interfaces.
Step 4	**lpts punt excessive-flow-trap penalty-rate** *protocol penalty_policer_rate* **Example:** `RP/0/RSP0/CPU0:router(config)# lpts punt excessive-flow-trap penalty-rate icmp 10`	Sets the penalty policing rate for a protocol. The penalty policer rate is in packets-per-second (pps) and ranges from 2 to 100. **Note** The penalty policing rate for a protocol consumes a policer rate profile.
Step 5	**lpts punt excessive-flow-trap penalty-timeout** *protocol time* **Example:** `RP/0/RSP0/CPU0:router(config)# lpts punt excessive-flow-trap penalty-timeout igmp 10`	Sets the penalty timeout value, which is a period of time that the interface trapped is placed in the penalty box, for a protocol. The penalty timeout value is in minutes and ranges from 1 to 1000. The default penalty timeout value is 15 minutes.
Step 6	**commit**	

Enabling Excessive Punt Flow Trap Processing: Examples

This is an example for enabling the Excessive Punt Flow Trap for subscriber interfaces, using the default penalty timeout (15 minutes) and setting a penalty rate of 20 pps for PPP and PPPoE protocols.

```
configure
lpts punt excessive-flow-trap subscriber-interfaces
lpts punt excessive-flow-trap penalty-rate ppp 20
lpts punt excessive-flow-trap penalty-rate pppoe 20
end
!!
```

This is an example for enabling the Excessive Punt Flow Trap for non-subscriber interfaces, using the default penalty rate (10 pps) and setting the ARP penalty timeout to 2 minutes.

```
configure
lpts punt excessive-flow-trap non-subscriber-interfaces
lpts punt excessive-flow-trap penalty-timeout arp 2
end
!!
```

Access Control List and Access Control List-based Forwarding

An Access Control List (ACL) is used to define access rights for a subscriber. It is also used for filtering content, blocking access to various network resources, and so on.

Certain service providers need to route certain traffic be routed through specific paths, instead of using the path computed by routing protocols. For example, a service provider may require that voice traffic traverse through certain expensive routes, but data traffic to use the regular routing path. This is achieved by specifying the next-hop address in the ACL configuration, which is then used for forwarding packet towards its destination. This feature of using ACL for packet forwarding is called ACL-based Forwarding (ABF).

The ACL is defined through CLI or XML; however, it can be applied to a subscriber session either through a dynamic-template, or through VSAs from RADIUS. Deploying ABF (using ACL) involves these stages:

- Defining an ACL, see Configuring Access-Control Lists, on page 226.
- Applying the ACL to an access-interface, see Activating ACL, on page 227.

Configuring Access-Control Lists

Perform this task to create an access control list. As an example, this access list is created to deploy ABF; therefore, it defines the next hop address.

SUMMARY STEPS

1. **configure**
2. {**ipv4** | **ipv6**} **access-list** *access-list-name*
3. *sequence-number* **permit** **tcp any any**
4. *sequence-number* **permit** {**ipv4** | **ipv6**} **host** *source_address* **nexthop** *source_address* *destination_address*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	{**ipv4** | **ipv6**} **access-list** *access-list-name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# ipv4 access-list foo_in`
	or
	`RP/0/RSP0/CPU0:router(config)# ipv6 access-list foo_in`
		Configures the access-list.

Step 3	*sequence-number* **permit** **tcp any any** **Example:** `RP/0/RSP0/CPU0:router(config)# 10 permit tcp any any`	Enters an access control list rule to tcp traffic.
Step 4	*sequence-number* **permit** {**ipv4** | **ipv6**} **host** *source_address* **nexthop** *source_address* *destination_address* **Example:**	Specifies packets to forward on ipv4 protocol from source IP address to destination IP address. **Note** Repeat steps 1 to 4 to configure the foo_out access-list.

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router(config)# 10 permit ipv4 host 9.8.8.9 nexthop 6.6.6.6 7.7.7.7`
	or
	`RP/0/RSP0/CPU0:router(config)# 10 permit ipv6 host 192:2:1:9 nexthop 192:2:6:8`	

Step 5	**commit**	

Configuring Access-Control Lists: Examples

```
//For IPv4
configure
ipv4 access-list foo_in
10 permit tcp any any
10 permit ipv4 host 9.8.8.9 nexthop 6.6.6.6 7.7.7.7
!
!
end
```

```
//For IPv6
configure
ipv6 access-list foo_in
10 permit tcp any any
10 permit ipv4 host 192:2:1:9 nexthop 192:2:6:8
!
!
end
```

Activating ACL

Perform this task to define a dynamic-template that is used to activate an access-control list.

SUMMARY STEPS

1. **configure**
2. **dynamic-template**
3. **type**{**ipsubscriber** | **ppp** | **service**} *dynamic-template-name*
4. {**ipv4** | **ipv6**} **access-group** *access-list-name* **ingress**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**dynamic-template** **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template`	Enters the dynamic-template configuration mode.
Step 3	**type** { **ipsubscriber** | **ppp** | **service** } *dynamic-template-name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type service foo`	Creates a service dynamic-template type.

Step 4	{**ipv4** | **ipv6**} **access-group** *access-list-name* **ingress**
	Example:
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv4 access-group foo_in ingress`
	or
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# ipv6 access-group foo_in ingress`
		Specifies access-control for the incoming packets.
		Note Similarly, create another access-group for the outgoing packets called foo_out.

Step 5	**commit**	

Activating ACL: Examples

```
//For IPv4
configure
dynamic-template
type service foo
ipv4 access-group foo_in ingress
!
!
end
```

```
//For IPv6
configure
dynamic-template
type service foo
ipv6 access-group foo_in ingress
!
!
end
```

Support for Lawful Intercept

Lawful Intercept allows Law Enforcement Agencies (LEAs) to conduct electronic surveillance as authorized by judicial or administrative order. Increasingly, legislation is being adopted and regulations are being enforced that require service providers (SPs) and Internet service providers (ISPs) to implement their networks to explicitly support authorized electronic surveillance. The types of SPs or ISPs that are subject to Lawful Intercept mandates vary greatly from country to country. Lawful Intercept compliance in the United States is specified by the Communications Assistance for Law Enforcement Act (CALEA).

Cisco ASR 9000 Series Router supports the Cisco Service Independent Intercept (SII) architecture and PacketCable™[1] Lawful Intercept architecture. The Lawful Intercept components by themselves do not ensure customer compliance with applicable regulations but rather provide tools that can be used by SPs and ISPs to construct an Lawful Intercept compliant network.

BNG supports the Per-session Lawful Intercept and Radius-based Lawful Intercept for subscribers. Both, per-session and radius-based lawful intercepts are executed on IPoE, PPPoE, and PPPoE LAC subscriber sessions in BNG.

Caution This guide does not address legal obligations for the implementation of lawful intercept. Service providers are responsible for ensuring that network complies with applicable lawful intercept statutes and regulations. It is recommended that legal advice be sought to determine obligations.

Note By default, Lawful Intercept is not a part of the Cisco IOS XR software. To enable Lawful Intercept, you must install and activate the **asr9k-li-px.pie**.

For more information about Lawful Intercept-related router configuration, see *Implementing Lawful Intercept* chapter in *Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide*.

Per-session Lawful Intercept

Lawful interception of all Layer 2 or Layer 3 traffic on a specified subscriber interface, on both ingress as well egress directions, and sending the replicated stream to mediation device, is called the per-session Lawful Intercept. This Lawful Intercept implements IPv4, IPv6, and multicast traffic interception using the Cisco-defined MIBs. By default, the SNMP-based Lawful Intercept feature is enabled on the Cisco ASR 9000 Series Router, which allows you to configure the taps. For more information about disabling SNMP-based Lawful Intercept, see Disabling SNMP-based Lawful Intercept, on page 230.

The subscriber session is identified by Account-session-ID, which acts as a key in identifying the specified subscriber interface for the subscriber user, whose traffic is getting intercepted.

Lawful Intercept, in general, can be implemented using either SII architecture or PacketCable™ specifications. The Cisco IOS-XR implementation of SNMP-based Lawful Intercept is based on service-independent intercept (SII) architecture. SNMPv3 authenticates data origin and ensures that the connection from Cisco ASR 9000 Series Router to the mediation device is secure. This ensures that unauthorized parties cannot forge an intercept target.

[1] PacketCable™ architecture addresses device interoperability and product compliance issues using the PacketCable™ Specifications.

Note To implement lawful intercept, you must understand how the SNMP server functions. For this reason, carefully review the information described in the module *Implementing SNMP* in *System Management Configuration Guide for Cisco ASR 9000 Series Routers*.

Lawful intercept must be explicitly disabled. It is automatically enabled on a provisioned router after installing and activating the **asr9k-li-px.pie**. However, you should not disable LI if there is an active tap in progress, because this deletes the tap.

Management plane must be configured to enable SNMPv3. Allows the management plane to accept SNMP commands, so that the commands go to the interface (preferably, a loopback) on the router. This allows the mediation device (MD) to communicate with a physical interface. For more information about Management Plane Protection feature, see Configuring the Inband Management Plane Protection Feature, on page 231 and for more information about enabling the mediation device, see Enabling the Mediation Device to Intercept VoIP and Data Sessions, on page 231.

Lawful Intercept MIBs

An external mediation device also known as collectors can create IPv4 or IPv6 address based TAPs using IP-TAP-MIB. The SNMPv3 protocol is used to provision the mediation device (defined by CISCO-TAP2-MIB) and the Taps(defined by CISCO-USER-CONNECTION-TAP-MIB). The Cisco ASR 9000 Series Router supports a total of 511 concurrent taps that includes both SNMP and Radius.

Lawful intercept uses these MIBs for interception:

- CISCO-TAP2-MIB—Used for lawful intercept processing. It contains SNMP management objects that control lawful intercepts on a Cisco ASR 9000 Series Router. The mediation device uses the MIB to configure and run lawful intercepts on targets sending traffic through the Cisco ASR 9000 Series Router. The CISCO-TAP2-MIB supports the SII feature and defines the provisioning of the mediation devices and generic Taps. It primarily consists of the mediation device table and a stream table. The mediation device table contains information about mediation devices with which the Cisco ASR 9000 Series Router communicates; for example, the device's address, the interfaces to send intercepted traffic over, and the protocol to use to transmit the intercepted traffic. The stream table contains a list of generic Taps that are provisioned by the MD table entries.
- CISCO-USER-CONNECTION-TAP-MIB—Used for intercepting traffic for individual subscribers. The MIB contains SNMP management objects to configure and execute wiretaps on individual user connections on the Cisco ASR 9000 Series Router. This MIB contains information about the user connections, each identified by a unique session ID. The CISCO-USER-CONNECTION-TAP-MIB cannot be configured without configuring the CISCO-TAP2-MIB.

Note It is not possible to configure an SNMP tap and a Radius tap at the same time. Also, the same session cannot be tapped more than once at a time.

Disabling SNMP-based Lawful Intercept

Lawful Intercept is enabled by default on the Cisco ASR 9000 Series Router after installing and activating the **asr9k-li-px.pie**.

- To disable Lawful Intercept, enter the **lawful-intercept disable** command in global configuration mode.

- To re-enable it, use the **no** form of this command.

Disabling SNMP-based Lawful Intercept: An example

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# lawful-intercept disable
```

Note The **lawful-intercept disable** command is available only after installing and activating the **asr9k-li-px.pie**.

All SNMP-based taps are dropped when lawful intercept is disabled.

Configuring the Inband Management Plane Protection Feature

If MPP was not earlier configured to work with another protocol, then ensure that the MPP feature is also not configured to enable the SNMP server to communicate with the mediation device for lawful interception. In such cases, MPP must be configured specifically as an inband interface to allow SNMP commands to be accepted by the router, using a specified interface or all interfaces.

Note Ensure this task is performed, even if you have recently migrated to Cisco IOS XR Software from Cisco IOS, and you had MPP configured for a given protocol.

For lawful intercept, a loopback interface is often the choice for SNMP messages. If you choose this interface type, you must include it in your inband management configuration.

Enabling the Mediation Device to Intercept VoIP and Data Sessions

These SNMP server configuration tasks enable the Cisco SII feature on a router running Cisco IOS XR Software by allowing the MD to intercept VoIP or data sessions.

SUMMARY STEPS

1. **configure**
2. **snmp-server view** *view-name* **ciscoTap2MIB included**
3. **snmp-server view** *view-name* **ciscoUserConnectionTapMIB included**
4. **snmp-server group** *group-name* **v3auth read** *view-name* **write** *view-name* **notify** *view-name*
5. **snmp-server host** *ip-address* **traps version 3 auth** *username* **udp-port** *port-number*
6. **snmp-server user** *mduser-id groupname* **v3 auth md5** md-*password*
7. **commit**
8. show snmp users
9. show snmp group
10. show snmp view

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**snmp-server view** *view-name* **ciscoTap2MIB included** **Example:** `RP/0//CPU0:router(config)# snmp-server view TapName ciscoTap2MIB included`	Creates or modifies a view record and includes the CISCO-TAP2-MIB family in the view. The SNMP management objects in the CISCO-TAP2-MIB that controls lawful intercepts are included. This MIB is used by the mediation device to configure and run lawful intercepts on targets sending traffic through the router.
Step 3	**snmp-server view** *view-name* **ciscoUserConnectionTapMIB included** **Example:** `RP/0//CPU0:router(config)# snmp-server view TapName ciscoUserConnectionTapMIB included`	Creates or modifies a view record and includes the CISCO-USER-CONNECTION-TAP-MIB family, to manage the Cisco intercept feature for user connections. This MIB is used along with the CISCO-TAP2-MIB to intercept and filter user traffic.
Step 4	**snmp-server group** *group-name* **v3auth read** *view-name* **write** *view-name* **notify** *view-name* **Example:** `RP/0//CPU0:router(config)# snmp-server group TapGroup v3 auth read TapView write TapView notify TapView`	Configures a new SNMP group that maps SNMP users to SNMP views. This group must have read, write, and notify privileges for the SNMP view.
Step 5	**snmp-server host** *ip-address* **traps version 3 auth** *username* **udp-port** *port-number* **Example:** `RP/0//CPU0:router(config)# snmp-server host 223.255.254.224 traps version 3 auth bgreen udp-port 2555`	Specifies SNMP trap notifications, the version of SNMP to use, the security level of the notifications, and the recipient (host) of the notifications.
Step 6	**snmp-server user** *mduser-id groupname* **v3 auth md5** md-*password* **Example:** `RP/0//CPU0:router(config)# snmp-server mduser-id TapGroup v3 auth md5 mdpassword`	Configures the MD user as part of an SNMP group, using the v3 security model and the HMAC MD5 algorithm, which you associate with the MD password. • The *mduser-id* and *mdpassword* must match that configured on MD. Alternatively, these values must match those in use on the router. • Passwords must be eight characters or longer to comply with SNMPv3 security minimums. • Minimum Lawful Intercept security level is auth; The noauth option will not work, as it indicates noAuthnoPriv security level. The Lawful Intercept security level must also match that of the MD. • Choices other than MD5 are available on the router, but the MD values must match.

	Command or Action	Purpose
		Most MDs default to or support only MD5.
Step 7	**commit**	
Step 8	show snmp users **Example:** `RP/0//CPU0:router# show snmp users`	Displays information about each SNMP username in the SNMP user table.
Step 9	show snmp group **Example:** `RP/0//CPU0:router# show snmp group`	Displays information about each SNMP group on the network.
Step 10	show snmp view **Example:** `RP/0//CPU0:router# show snmp view`	Displays information about the configured views, including the associated MIB view family name, storage type, and status.

Enabling the Mediation Device to Intercept VoIP and Data Sessions: An example

```
configure
snmp-server view TapName ciscoTap2MIB included
snmp-server view TapName ciscoUserConnectionTapMIB included
snmp-server group TapGroup v3 auth read TapView write TapView notify TapView
snmp-server host 223.255.254.224 traps version 3 auth bgreen udp-port 2555
snmp-server mduser-id TapGroup v3 auth md5 mdpassword
end
!
!
```

Radius-based Lawful Intercept

Radius-based Lawful Intercept feature provides mechanisms for interception of the BNG subscriber traffic by using the RADIUS attributes. This is a preferred method over SNMP user-connection MIB, as SNMP-based method prevents a session to be tapped until an IP address has been assigned to the session. In the Radius-based LI mechanism, tapping is possible as soon as a session is established.

A RADIUS-based Lawful Intercept solution enables intercept requests to be sent (through Access-Accept packets or Change of Authorization (CoA)-Request packets) to the network access server (NAS) or to the Layer 2 Tunnel Protocol access concentrator (LAC) from the RADIUS server. All traffic data going to or from a PPP or L2TP session is passed to a mediation device. Another advantage of RADIUS-based Lawful Intercept solution is to set the tap with Access-Accept packets that allows all target traffic to be intercepted simultaneously.

The RADIUS-based Lawful Intercept feature provides tap initiation support for these modes:

- Access-Accept based Lawful Intercept for the new session
- CoA based Lawful Intercept for existing session

Note By default, the Radius-based Lawful Intercept functionality is not enabled. For more information about enabling Radius-based Lawful Intercept, see Enabling RADIUS-based Lawful Intercept, on page 234.

Enabling RADIUS-based Lawful Intercept

Perform this task to enable the Radius-based Lawful Intercept feature.

SUMMARY STEPS

1. **configure**
2. **aaa intercept**
3. **aaa server radius dynamic-author**
4. **port** *port_number*
5. **server-key** [**0**|**7**] *word*
6. **client** *hostname*{ **vrf** *vrf_name* | **server-key** [**0**|**7**] *word* }
7. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**aaa intercept** **Example:** `RP/0/RSP0/CPU0:router(config)# aaa intercept`	Enables the radius-based lawful intercept feature. **Note** This command is available only after installing and activating **asr9k-li-px.pie**. When you disable aaa intercept, all radius-based taps are removed from the Cisco ASR 9000 Series Router.
Step 3	**aaa server radius dynamic-author** **Example:** `RP/0/RSP0/CPU0:router(config)# aaa server radius dynamic-author`	Configures the lawful intercept as a AAA server and enters the dynamic authorization local server configuration mode.
Step 4	**port** *port_number* **Example:** `RP/0/RSP0/CPU0:router(config-Dynamic Author)# port 1600`	Specifies the RADIUS server port. The default port number is 1700.
Step 5	**server-key** [**0**|**7**] *word* **Example:** `RP/0/RSP0/CPU0:router(config-Dynamic Author)# server-key cisco`	Specifies the encryption key shared with the RADIUS client.
Step 6	**client** *hostname*{ **vrf** *vrf_name* | **server-key** [**0**|**7**] *word* }	Specifies the client with which the AAA server will be communicating.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router(config-Dynamic Author)# client 3.0.0.28 vrf default server-key cisco`	**Note** You can configure the server key in a global mode and also as a per client type key.
Step 7	**commit**	

Enabling RADIUS-based Lawful Intercept: An example

```
configure
aaa intercept
aaa server radius dynamic-author
port 1600
server-key cisco
client 3.0.0.28 vrf default server-key cisco
end
!
!
```

What to do next

These attributes need to be present in the user profile to configure the Radius-based Lawful Intercept.

```
xyz_user1@domain.com Password == "cisco"
    Cisco-avpair = "md-ip-addr=192.1.1.4",
    Cisco-avpair += "md-port=203",
    Cisco-avpair += "md-dscp=3",
    Cisco-avpair += "intercept-id=abcd0003",
    Cisco-avpair += "li-action=1"
```

TCP MSS Adjustment

The TCP MSS Adjustment feature allows the configuration of the maximum segment size (MSS) on transient packets that traverse a Cisco ASR 9000 Series Router.

When dealing with PPPoE or L2TP cases, an additional header that the client initiating a TCP session may not be aware of is added to the packet. This can result in lost packets, broken transmissions, or fragmentation when packet sizes exceed the maximum transmission units (MTUs) due to the added headers.

Here is a sample scenario that shows how the TCP MSS adjust feature works:

Figure 17: Sample TCP MSS Adjust

In this example, the HTTP client sends to the HTTP server a TCP synchronize (SYN) packet that signals an MSS value of 1300 (MTU) - 20 TCP - 20 IP header = 1260. On receiving it, the HTTP server acknowledges it with a SYN ACK message. The HTTP client confirms the TCP session with a single acknowledgment and opens up the TCP channel.

Note This is a sample scenario without PPPoE or L2TP.

When the HTTP server picks up a large file, it segments it into 1460 byte chunks (assuming that there are no http headers for now). When the HTTP server sends the packet, the first Cisco ASR 9000 Series Router (on the right) detects that the MTU is 576 downstream to the client and requires a 1300 byte packet to be fragmented.

If the server sets the DF ("don't fragment") bit, then the packet is dropped. And, if the packet does not have the DF bit set, then it gets fragmented, requiring the client to reassemble the packets. In digital subscriber line (DSL) or fibre-to-the-home (FTTH) like access, a CPE may block incoming fragments as a security mechanism, causing this transmission to be lost.

In a typical scenario, having packets that are dropped causes partial downloads, an obstruction, or a delay in displaying images in web pages. MSS adjust overcomes this scenario by intercepting the TCP SYN packet, reading the MSS option, and adjusting the value so that the server does not send packets larger than the configured size (plus headers).

Note that the TCP MSS value is only adjusted downward. If the clients request an MSS value lower than the configured value, then no action is taken.

In the case of PPPoE, an extra 8 bytes and in the case of L2TP, an extra 40 bytes is added to the packet. The recommended MSS adjust values are 1452 for PPPoE, and 1420 for L2TP scenarios, assuming a minimum MTU of 1500 end-to-end.

Separate unique global values for PTA and L2TP are supported, which once configured allows all future sessions to be TCP MSS adjustment; however, the sessions already established will not be TCP adjusted. If the global value is changed, then all new TCP subscriber sessions, will get the new global value.

For more information about configuring the TCP MSS value of packets, see Configuring the TCP MSS Value of TCP Packets, on page 237.

Note To disable this on a session, you must first disable the global configuration, then delete the session and recreate it.

TCP encapsulated in both IPv4 and IPv6 are supported.

Restrictions

These restrictions are applicable for TCP MSS Adjustment:

- Because the MSS is TCP-specific, the TCP MSS Adjustment feature is applicable only to (transit) TCP packets and the UDP packets are unaffected.
- TCP MSS Adjustment configuration affects only the PPPoE PTA and LAC sessions types. It does not affect IP sessions or any non-BNG interfaces.
- The MSS option must be the first option in the TCP header.
- The router uses the MSS value that the user configures for checking TCP/IPV4 packets. When checking TCP/IPv6 packets, the router automatically adjusts the configured MSS value down by 20 bytes to account for the larger IPv6 header. For example, if the TCP MSS value is configured to 1450, then the router adjusts the TCP MSS in an IPV4 packet down to 1450 and down to 1430 for an IPv6 packet.

Configuring the TCP MSS Value of TCP Packets

Perform this task to configure the TCP MSS value of TCP packets in order to prevent TCP sessions from being dropped.

SUMMARY STEPS

1. **configure**
2. **subscriber**
3. **pta tcp mss-adjust** *max-segment-size*
4. **commit**
5. **configure**
6. **vpdn**
7. **l2tp tcp-mss-adjust** *max-segment-size*
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**subscriber** **Example:** `RP/0/RSP0/CPU0:router(config)# subscriber`	Enables the subscriber configuration mode.
Step 3	**pta tcp mss-adjust** *max-segment-size* **Example:** `RP/0/RSP0/CPU0:router(config-subscriber)# pta tcp mss-adjust 1300`	Sets the MSS value of TCP packets going through a Cisco ASR 9000 Series Router for a PTA subscriber. The TCP MSS Adjust maximum segment size ranges from 1280 to 1536 (in bytes). **Note** The value represents the global value for the PTA sessions, when the feature is enabled it applies to all sessions.
Step 4	**commit**	
Step 5	**configure**	
Step 6	**vpdn** **Example:** `RP/0/RSP0/CPU0:router(config)# vpdn`	Enables the vpdn configuration mode.
Step 7	**l2tp tcp-mss-adjust** *max-segment-size* **Example:** `RP/0/RSP0/CPU0:router(config-vpdn)# l2tp tcp-mss-adjust 1300`	Sets the MSS value of TCP packets going through a Cisco ASR 9000 Series Router for a LAC subscriber. The TCP MSS Adjust maximum segment size ranges from 1280 to 1460 (in bytes).
Step 8	**commit**	

Configuring the TCP MSS Value of TCP Packets: Examples

```
//Example for configuring the TCP MSS value of TCP packets for a PPPoE PTA subscriber
session:

configure
subscriber
pta tcp mss-adjust 1280
!!


// Example for configuring the TCP MSS value of TCP packets for a PPPoE LAC subscriber
session:

configure
vpdn
l2tp tcp-mss-adjust 1460
!!
```

Linking to Subscriber Traffic in a Shared Policy Instance Group

You can associate the subscriber traffic belonging to a Shared Policy Instance (SPI) group of multiple subinterfaces with a link using a Cisco Vendor-Specific Attribute (VSA). When you apply member hash Cisco:Avpair from RADIUS for a SPI group, traffic for that group will not spill across members. You can identify hash to member mapping based on the bundle's Link Ordering Number (LON).

To enable this feature, configure the following Cisco VSA in the RADIUS profile of the subscriber:

```
Cisco-avpair = "subscriber:member-hash=XX"
```

where XX is the hash value.

Supported Features

- IPoE and PPPoE call flows
- IPv4 and IPv6
- Member hash can be downloaded from RADIUS server
- Traffic is programmed when a new hash value is downloaded and also when a bundle member is modified
- High availability scenarios such as Flap, LC OIR, Process restart, and RPFO
- Only route processor subscribers and with maximum scale

Verifying Hash Value

To display the hash value programmed for the subscriber session, refer to Flow-tag value in the **show route** *address* **detail** command output:

```
RP/0/0/CPU0:server#show route 10.0.0.1/32 detail
Mon Mar  2 20:08:29.079 IST

Routing entry for 10.0.0.1/32
  Known via "subscriber", distance 2, metric 0 (connected)
  Installed Mar  2 20:07:35.448 for 00:00:54
```

```
Routing Descriptor Blocks
    directly connected, via GigabitEthernet0/0/0/0.pppoe1
      Route metric is 0
      Label: 0x300 (768)
      Tunnel ID: None
      Extended communities count: 0
      NHID:0x0(Ref:0)
  Route version is 0x1 (1)
  No local label
  IP Precedence: Not Set
  QoS Group ID: Not Set
  Flow-tag: 33
  Route Priority: RIB_PRIORITY_RECURSIVE (9) SVD Type RIB_SVD_TYPE_LOCAL
  Download Priority 3, Download Version 5
  No advertising protos.
```

Subscriber Session on Ambiguous VLANs

Ambiguous VLAN enables you to create multiple subscriber sessions on a single access-interfaces. As a result, it increases the scalability of the access-interface. An ambiguous VLAN is an L3 interface on which either a VLAN ID range, or a group of individual VLAN IDs are specified. Instead of individually mapping each subscriber to a VLAN, an ambiguous VLAN configuration performs the mapping for a group. Multiple subscribers can be mapped on the ambiguous VLAN as long as they possess a unique MAC address. The subscriber sessions created over ambiguous VLANs are identical to the ones created over regular VLANs, and support all regular configurations such as policy-map, VRFs, QoS, access-control list, and so on.

For enabling IPoE subscriber session creation on an ambiguous VLAN, see Establishing Subscriber Session on Ambiguous VLANs, on page 239.

From Cisco IOS XR Release 5.1.3 and later, the DHCP offer can be send as Unicast (or as per the broadcast policy flag in the DHCP request) for ambiguous VLANs. The ambiguous VLAN configuration in this case, must use a range of VLAN tags (For example, **encapsulation ambiguous dot1q** *10, 100*).

For ambiguous VLAN dot1q configuration where the match criteria is explicitly configured for inner and outer VLAN tags or where a range is specified or where **any** is used for outer VLAN tag, the MTU is calculated by adding 8 bytes (2xdot1q tags) to the default MTU. That is, if default is 1514, the MTU is set to 1522 bytes in such scenarios. Whereas, for configurations where the match criteria for inner VLAN is specified as **any**, the MTU on the sub-interface is calculated by adding 4 (and not 8) bytes to the main interface MTU. That is, 1514 + 4 = 1518 bytes. This behavior is applicable for both physical interfaces and bundle sub-interfaces.

Restriction

The use of **any** tag in the ambiguous VLAN configuration is not supported for Unicast DHCP offers. The DHCP offer packets are not forwarded to the subscriber if **any** tag is used in the configuration.

A DHCP proxy debug error message saying, ARP is not supported on ambiguous VLAN interface, is logged in such failure scenarios.

Establishing Subscriber Session on Ambiguous VLANs

Perform this task to define an ambiguous VLAN and enable creation of IP subscriber session on it.

Note There is no DHCP-specific configuration required for ambiguous VLANs.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. Use any of these commands to configure encapsulated ambiguous VLANs:
   - **encapsulation ambiguous** { **dot1q** | **dot1ad** } {**any** | *vlan-range* }
   - **encapsulation ambiguous dot1q** *vlan-id* **second-dot1q** { **any** | *vlan-range* }
   - **encapsulation ambiguous dot1q any second-dot1q** { **any** | *vlan-id* }
   - **encapsulation ambiguous dot1ad** *vlan-id* **dot1q** { **any** | *vlan-range* }
   - **encapsulation ambiguous dot1q** *vlan-range* **second-dot1q any**
   - **encapsulation ambiguous dot1ad** *vlan-range* **dot1q any**
4. **ipv4 | ipv6address** *source-ip-address destination-ip-address*
5. **service-policy type control subscriber** *policy_name*
6. **ipsubscriber ipv4 l2-connected**
7. **initiator dhcp**
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/1/0/0.12`	Configures the interface and enters the interface configuration mode.
Step 3	Use any of these commands to configure encapsulated ambiguous VLANs: • **encapsulation ambiguous** { **dot1q** | **dot1ad** } {**any** | *vlan-range* } • **encapsulation ambiguous dot1q** *vlan-id* **second-dot1q** { **any** | *vlan-range* } • **encapsulation ambiguous dot1q any second-dot1q** { **any** | *vlan-id* } • **encapsulation ambiguous dot1ad** *vlan-id* **dot1q** { **any** | *vlan-range* } • **encapsulation ambiguous dot1q** *vlan-range* **second-dot1q any** • **encapsulation ambiguous dot1ad** *vlan-range* **dot1q any** **Example:**	Configures IEEE 802.1Q VLAN configuration. The *vlan-range* is given in comma-separated, or hyphen-separated format, or a combination of both, as shown in the examples. **Note** Although **encapsulation ambiguous dot1ad** is supported, it is not commonly used in BNG deployments. **encapsulation ambiguous dot1q any** is not supported for unicast DHCP offers. You must use **encapsulation ambiguous dot1q** *vlan-range* for such scenarios.

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q any`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q 14 second-dot1q 100-200`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q any second-dot1q any`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1ad 14 dot1q 100,200,300-400`
	`RP/0/RSP0/CPU0:router(config-if)# encapsulation ambiguous dot1q 1-1000 second-dot1q any`	

Step 4	**ipv4 | ipv6address** *source-ip-address destination-ip-address*
	Example:
	`RP/0/RSP0/CPU0:router(config-if)# ipv4 address 2.1.12.1 255.255.255.0`
	`RP/0/RSP0/CPU0:router(config-if)# ipv6 address 1:2:3::4 128`
		Configures the IPv4 or IPv6 protocol address.

Step 5	**service-policy type control subscriber** *policy_name* **Example:** `RP/0/RSP0/CPU0:router(config-if)# service-policy type control subscriber PL1`	Applies a policy-map to an access interface where the policy-map was previously defined with the specified PL1 policy_name.
Step 6	**ipsubscriber ipv4 l2-connected** **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipsubscriber ipv4 l2-connected`	Enables l2-connected IPv4 IP subscriber.
Step 7	**initiator dhcp** **Example:** `RP/0/RSP0/CPU0:router(config-if)# initiator dhcp`	Enables initiator DHCP on the IP subscriber.
Step 8	**commit**	

Establishing Subscriber Session on Ambiguous VLANs: An example

```
configure
interface Bundle-Ether100.10
encapsulation ambiguous dot1q 14 second-dot1q any
ipv4 address 2.1.12.12 55.255.255.0
service-policy type control subscriber PL1
ipsubscriber ipv4 l2-connected
!
!
end
```

Outer VLAN Range

The Outer VLAN range is a BNG-specific feature that provides a more advanced VLAN encapsulation option of double-tagged VLANs, where the outer VLAN is specified as a range and the inner VLAN is specified as **any**.

The current BNG implementation supports a high scale of subscriber interface. However, due to QoS hardware limitation, the number of subscribers with QoS policies attached under a single L3 ambiguous VLAN sub-interface is limited to 8K. Therefore, in a large scale scenario, if QoS policies are to be attached to each of the subscribers and if the maximum scale per port is to be achieved, you must configure multiple L3 ambiguous VLAN sub-interfaces per port, with encapsulations that partition the subscribers among the VLAN sub-interfaces. The encapsulations used in such scenarios are:

- Single-tagged VLAN range encapsulations.
- Double-tagged encapsulation, with an inner VLAN range.
- Double-tagged encapsulations, with a fixed outer VLAN-ID and an inner VLAN match for **any**.

In certain scenarios, depending on how the VLAN-IDs are allocated for the subscribers, none of the above partitioning schemes may be suitable. In such scenarios, the L3 ambiguous encapsulation double tag that matches an outer VLAN range and **any** inner VLAN can be used.

The configuration options available for the Outer VLAN range feature are:

- **encapsulation ambiguous dot1q** *vlan range* **second-dot1q any**
- **encapsulation ambiguous dot1ad** *vlan range* **dot1q any**

Sample Configuration for Outer VLAN Range

The sample configuration listed in this section shows how to configure 32K subscribers for each physical interface, using a double-tagged encapsulation to partition the subscribers across four sub-interfaces. Here, 8K subscribers, each with a separate QoS policy applied, are configured for each VLAN sub-interface. Further, a total of four VLAN sub-interfaces are configured to support 32K subscribers for each physical interface.

Option 1: Four VLAN sub-interfaces

```
interface GigabitEthernet0/0/0/0.1
encapsulation ambiguous dot1q 1-1000 second-dot1q any
!
interface GigabitEthernet0/0/0/0.2
encapsulation ambiguous dot1q 1001-2000 second-dot1q any
!
interface GigabitEthernet0/0/0/0.3
encapsulation ambiguous dot1q 2001-3000 second-dot1q any
!
interface GigabitEthernet0/0/0/0.4
encapsulation ambiguous dot1q 3001-4000 second-dot1q any
!
```

Option 2: Nine VLAN configuration ranges

```
interface GigabitEthernet0/0/0/0.1
encapsulation ambiguous dot1q 9-18, 19-25, 26, 27-30, 32, 33-40, 42, 43-50, 52 second-dot1q
 any
```

```
!
```

Verification of Outer VLAN Range Configurations

These show commands can be used to verify the outer VLAN range configurations in BNG:

SUMMARY STEPS

1. **show interface***VLAN sub-interface*
2. **show ethernet tags** *VLAN sub-interface*
3. **show ethernet tags** *VLAN sub-interface* **detail**

DETAILED STEPS

Step 1 **show interface***VLAN sub-interface*

Displays VLAN sub-interface details, including encapsulations.

Example:

```
RP/0/RSP0/CPU0:router#
show interfaces GigabitEthernet 0/1/0/10.12
GigabitEthernet0/1/0/10.12 is up, line protocol is up
  Interface state transitions: 1
  Hardware is VLAN sub-interface(s), address is 0022.bde2.b222
  Internet address is Unknown
  MTU 1518 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation 802.1Q Virtual LAN,
    Outer Match: Dot1Q VLAN 11-20,21-30,31-60,61-100,101-140,141-180,181-220,221-260,261-300
    Inner Match: Dot1Q VLAN any
    Ethertype Any, MAC Match src any, dest any
  loopback not set,
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
- - - - -
- - - - -
```

Step 2 **show ethernet tags** *VLAN sub-interface*

Displays VLAN sub-interface outer tag information, including outer VLAN ranges.

Example:

```
RP/0/RSP0/CPU0:router#
show ethernet tags tengigE 0/0/0/0.1
St:    AD - Administratively Down, Dn - Down, Up - Up
Ly:    L2 - Switched layer 2 service, L3 = Terminated layer 3 service,
Xtra   C - Match on Cos, E  - Match on Ethertype, M - Match on source MAC
-,+:   Ingress rewrite operation; number of tags to pop and push respectively

Interface               St  MTU  Ly Outer            Inner            Xtra -,+
Te0/0/0/0.1             Up  1522 L3 .1Q:10           .1Q:100-200      -    0 0
- - - - -
- - - - -
```

Step 3 **show ethernet tags** *VLAN sub-interface* **detail**

Displays VLAN sub-interface outer tag information, including outer VLAN ranges, in detail.

Example:

```
RP/0/RSP0/CPU0:router#
show ethernet tags  GigabitEthernet 0/0/0/0.1 detail
GigabitEthernet0/1/0/10.12 is up, service is L3
    Interface MTU is 1518
    Outer Match: Dot1Q VLAN 11-20,21-30,31-60,61-100,101-140,141-180,181-220,221-260,261-300
    Inner Match: Dot1Q VLAN any
    Local traffic encap: -
    Pop 0 tags, push none
```

Limitations of Outer VLAN Range

The Outer VLAN Range feature is subjected to these restrictions:

- It is specific to BNG.
- The double-tagged L3 ambiguous encapsulation that matches an outer VLAN range and **any** inner VLAN, and an overlapping single tag encapsulation must not be configured at the same time under the same parent trunk interface. For example, the configurations listed here shows a double-tagged encapsulation configured under one sub-interface and a single-tagged encapsulation configured under another sub-interface of the same parent interface. Although it is not a valid configuration, the system does not reject it.

```
interface Bundle-ether 1.1
encapsulation ambiguous dot1q 2-100 second any
!
interface Bundle-ether 1.2
encapsulation ambiguous dot1q 3
```

- Network layer protocols must not be configured on L3 VLAN sub-interfaces configured with VLAN ranges or the **any** keyword. If they are configured in that manner, then any layer 3 traffic may be dropped. This is a limitation of generic ambiguous VLANs, and is applicable to BNG-specific outer VLAN range feature too.

uRPF

Unicast Reverse Path Forwarding (uRPF) is a feature in BNG that verifies whether the packets that are received on a subscriber interface are sent from a valid subscriber. uRPF only applies to subscribers using an L3 service.

For PPPoE subscribers, the uRPF check ensures that the source address in the arriving packet matches the set of addresses associated with the subscriber. The subscriber addresses are the IPCP assigned addresses, or any framed routed assigned through RADIUS. PPPoE subscribers are identified by session ID and VLAN keys. BNG performs the uRPF check to ensure that the source IP address in the arriving packets matches the expected session IDs and VLAN keys.

For IPoE subscribers, the subscriber addresses are the ones assigned through DHCP. IPoE subscribers are identified by the incoming MAC address. The uRPF check ensures that the source IP address is the one allocated by DHCP to the source MAC address.

uRPF is supported on both IPv4 and IPv6 subscribers and is enabled using a dynamic template. To define a dynamic template for enabling uRPF, see Creating Dynamic Template for IPv4 or IPv6 Subscriber Session, on page 77.

Multicast Services

Multicast services enable multiple subscribers to be recipients of a single transmission from one source. For example, real-time audio and video conferencing makes good use of a multicast service. The multicast features applied on the PPPoE interfaces of BNG includes:

Multicast Coexistence

On BNG, the multicast services coexist with regular unicast services. The multicast feature on BNG is the same as the existing L3 multicast feature already supported on the Cisco ASR 9000 Series Routers. On BNG, multicast is enabled on the trunk interfaces, and the VLANs created over physical interfaces and bundles. Multicast co-existence works for PPPoE PTA subscriber sessions. For more details on multicast implementation on ASR9k, see *Implementing Layer-3 Multicast Routing on Cisco IOS XR Software* chapter in *Multicast Configuration Guide for Cisco ASR 9000 Series Routers*.

To enable multicast function on BNG, see Enabling Address Family for the VRF, on page 245.

Enabling Address Family for the VRF

Perform this task to enable multicast functions for the required address family.

SUMMARY STEPS

1. **configure**
2. **multicast-routing**
3. **vrf** *vrf_name*
4. **address-family ipv4**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**multicast-routing** **Example:** `RP/0/RSP0/CPU0:router(config)# multicast routing`	Configures multicast-routing.
Step 3	**vrf** *vrf_name* **Example:** `RP/0/RSP0/CPU0:router(config)# vrf vrf1`	Configures the vrf name.

	Command or Action	Purpose
Step 4	**address-family ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# address-family ipv4`	Enables the multicast functions in the ipv4 address family.
Step 5	**commit**	

Enabling Address Family for the VRF: An example

```
multicast-routing
vrf vrf1
address-family ipv4
 !
!
end
```

Multicast Replication

BNG supports the multicast packet replication on PPPoE interfaces. It also supports multicast forwarding on subscriber interfaces, and transmission of multicast IP video content. When the multicast replication is enabled for a subscriber, BNG performs IGMP statistics gathering for that subscriber, and has the ability to export them. Multicast replication is supported on subscriber interfaces, which are configured in the passive mode.

HQoS Correlation

The Hierarchical quality of service (HQoS) correlation feature monitors every subscriber's multicast bandwidth usage through IGMP reports received on each subscriber's PPPoE session, and limits the unicast bandwidth usage, to leave enough bandwidth for multicast traffic. This is useful when the multicast traffic and unicast traffic share the same physical link to the subscriber in the last mile, when the multicast and unicast traffic are forwarded onto the last mile link by different devices. This feature is configured on BNG that forwards the unicast traffic to the subscriber. Based on the IGMP reports received, BNG informs the unicast QoS shaper on the PPPoE session to alter the bandwidth limit allowed for unicast traffic flows. Using this HQoS correlation feature, a service provider can protect the multicast traffic to the PPPoE subscriber from bursty unicast traffic. The bandwidth profiles for multicast flows need to be configured on BNG.

To define the bandwidth profile, see Configuring Minimum Unicast Bandwidth, on page 246.

To specify the mode for Multicast HQoS, see Configuring Multicast HQOS Correlation Mode or Passive Mode, on page 248.

Configuring Minimum Unicast Bandwidth

A minimum unicast bandwidth can be configured, to prevent unicast traffic from being completely cut off by oversubscribed multicast traffic. Perform this task to set the guaranteed minimum unicast bandwidth for a subscriber using QoS.

SUMMARY STEPS

1. **configure**
2. **dynamic-template**

3. **type** [**ppp** | **ip-subscriber** | **service**] *name*
4. **qos output minimum-bandwidth** *range*
5. **exit**
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dynamic-template** **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template`	Enters dynamic template configuration mode.
Step 3	**type** [**ppp** | **ip-subscriber** | **service**] *name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ppp p1`	. Specifies the type of dynamic template that needs to be applied. Three available types are: • PPP • IP-subscriber • Service
Step 4	**qos output minimum-bandwidth** *range* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# qos output minimum-bandwidth 10`	Sets the guaranteed minimum bandwidth, in kbps, for a subscriber. Range is from 1 to 4294967295.
Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# exit`	Exits from the current mode.
Step 6	**commit**	

Configuring Minimum Bandwidth: An example

```
configure
dynamic-template
type ppp p1
service-policy output pmap
multicast ipv4 qos-correlation
qos output minimum-bandwidth 10
end
```

Configuring Multicast HQOS Correlation Mode or Passive Mode

Perform this task to configure multicast in HQoS correlation mode or passive mode to enable multicast replication over PPPoE interfaces.

SUMMARY STEPS

1. **configure**
2. **dynamic-template**
3. **type ppp** *dynamic-template name*
4. **multicast ipv4 <qos-correlation | passive>**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dynamic-template** **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template`	Enter the dynamic-template configuration mode.
Step 3	**type ppp** *dynamic-template name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ppp foo`	Enters the ppp type mode to configure igmp for subscriber interfaces.
Step 4	**multicast ipv4 <qos-correlation | passive>** **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# multicast ipv4 qos-correlation`	Configures the subscriber either in the QoS-correlation mode (IGMP-HQOS correlation), or passive mode (multicast forwarding).
Step 5	**commit**	

Configuring Multicast HQOS Correlation Mode: An example

```
dynamic-template type ppp foo
multicast ipv4 qos-correlation
 !
!
end
```

IGMP to Unicast QoS Shaper Correlation

The Unicast QoS Shaper correlation feature configures the bandwidth profiles for the multicast flows and allows the IGMP messages to derive the multicast bandwidth usage for each subscriber. On the PPPoE subscriber sessions, the amount of multicast bandwidth that a subscriber uses is deducted from the unicast QoS shaper until a minimum threshold is reached.

For more information about configuring the IGMP QoS shaper, see Configuring the IGMP to HQoS Correlation Feature in a VRF, on page 249. For more information about configuring the IGMP for subscriber interfaces, see Configuring IGMP Parameters for Subscriber Interfaces, on page 251.

IGMP uses route-policies to distribute the absolute rate for all multicast flows. For more information for configuring the route-policy for unicast QoS shaper, see Configuring route-policy for Unicast QoS Shaper, on page 250.

Configuring the IGMP to HQoS Correlation Feature in a VRF

Perform this task to configure the IGMP to HQoS Correlation Feature in a VRF.

SUMMARY STEPS

1. **configure**
2. **router igmp**
3. **unicast-qos-adjust adjustment-delay** *time*
4. **unicast-qos-adjust download-interval** *time*
5. **unicast-qos-adjust holdoff** *time*
6. **vrf** *vrf-name*
7. **traffic profile** *profile-name*
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router igmp** **Example:** `RP/0/RSP0/CPU0:router(config)# router igmp`	Enters the router process for IGMP configuration mode.
Step 3	**unicast-qos-adjust adjustment-delay** *time* **Example:** `RP/0/RSP0/CPU0:router(config-igmp)# unicast-qos-adjust adjustment-delay 1`	Configures the time to wait before programming rate in IGMP QoS shaper for subscriber unicast traffic. The time to wait ranges from 0 to 10 seconds.
Step 4	**unicast-qos-adjust download-interval** *time* **Example:** `RP/0/RSP0/CPU0:router(config-igmp)# unicast-qos-adjust download-interval 10`	Configures the time before downloading a batch of interfaces to IGMP QoS shaper for subscriber unicast traffic. The download interval time ranges from 10 to 500 milliseconds.
Step 5	**unicast-qos-adjust holdoff** *time* **Example:** `RP/0/RSP0/CPU0:router(config-igmp)# unicast-qos-adjust holdoff 5`	Configures the hold-off time before QoS clears the stale entries for the IGMP QoS shaper. The hold-off time ranges from 5 to 1800 seconds.
Step 6	**vrf** *vrf-name* **Example:**	Enters the VRF configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-igmp)# vrf vrf1`	
Step 7	**traffic profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config-igmp-vrf1)# traffic profile routepolicy1`	Configures the route-policy to be used to map the bandwidth profile.
Step 8	**commit**	

Configuring the IGMP QoS Shaper: An Example

```
configure
router igmp
unicast-qos-adjust adjustment-delay 1
unicast-qos-adjust download-interval 10
unicast-qos-adjust holdoff 5
vrf vrf1
traffic profile routepolicy1
!
!
end
```

Configuring route-policy for Unicast QoS Shaper

Perform this task to configure route-policy for unicast QoS shaper.

SUMMARY STEPS

1. **configure**
2. **router igmp**
3. **vrf** *vrf-name*
4. **traffic profile** *profile-name*
5. **commit**
6. **show igmp unicast-qos-adjust statistics**
7. **show igmp unicast-qos-adjust statistics interface** *interface-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router igmp** **Example:** `RP/0/RSP0/CPU0:router(config)# router igmp`	Enter the router process for igmp configuration mode.
Step 3	**vrf** *vrf-name* **Example:**	Enters the vrf configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-igmp)# vrf vrf1`	
Step 4	**traffic profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config-igmp-vrf1)# traffic profile routepolicy1`	Configures the route-policy to be used to map the bandwidth profile.
Step 5	**commit**	
Step 6	**show igmp unicast-qos-adjust statistics** **Example:** `RP/0/RSP0/CPU0:router# show igmp unicast-qos-adjusted statistics`	(Optional) Displays the internal statistics of the feature, such as total number of interface groups under adjustment, uptime since last clear command, and total number of rate adjustment calls for unicast QoS shaper.
Step 7	**show igmp unicast-qos-adjust statistics interface** *interface-name* **Example:** `RP/0/RSP0/CPU0:router# show igmp unicast-qos-adjusted statistics interface interface1`	(Optional) Displays the interface name, number of flows adjusted, total rate adjusted, uptime after first adjustment for unicast QoS shaper.

Configuring route-policy for Unicast QoS Shaper: Examples

```
#Adding a route-policy for profile1

route-policy profile1
if destination in (239.0.0.0/8 le 32) then
set weight 1000
endif
end-policy

# Configuring profile1 for Unicast QoS Shaper
router igmp
vrf vrf1
traffic profile profile1
 !
!
end
```

Configuring IGMP Parameters for Subscriber Interfaces

Perform this task to configure IGMP parameters for subscriber interfaces.

SUMMARY STEPS

1. **configure**
2. **dynamic-template**
3. **type ppp** *dynamic-template name*
4. **igmp explicit-tracking**
5. **igmp query-interval** *value*
6. **igmp query-max-response-time** *query-response-value*

7. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dynamic-template** **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template`	Enter the dynamic-template configuration mode.
Step 3	**type ppp** *dynamic-template name* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template)# type ppp foo`	Enters the ppp type mode to configure igmp for subscriber interfaces.
Step 4	**igmp explicit-tracking** **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# igmp explicit-tracking`	Enables IGMPv3 explicit host tracking.
Step 5	**igmp query-interval** *value* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# igmp query-interval 60`	Sets the query-interval in seconds for igmp. **Note** The igmp query-interval value, in seconds, should be in the range from 1 to 3600. With 16000 PPPoE subscribers or less, the recommended value, that also the default, is 60 seconds.
Step 6	**igmp query-max-response-time** *query-response-value* **Example:** `RP/0/RSP0/CPU0:router(config-dynamic-template-type)# igmp query-max-response-time 4`	Sets the query-max-response-time in seconds for igmp. **Note** The igmp query-interval value, in seconds, is in the range from 1 to 12.
Step 7	**commit**	

Configuring IGMP for Subscriber Interfaces: An example

```
dynamic-template type ppp foo
igmp explicit-tracking
igmp query-interval 60
igmp query-max-response-time 4
 !
!
end
```

IGMP Accounting

The Internet Group Management Protocol (IGMP) accounting feature enables BNG to maintain a statistics file to log the instances of subscriber joining, or leaving a multicast group. The file's format is:

```
harddisk:/usr/data/igmp/accounting.dat.<Node ID>.<YYMMDD>
```

where

- Node ID is the name of the node that generates the file; for example, RP/0/RSP0/CPU0.
- YY is the year, MM is the month, and DD is the day.

An example of the statistics file name is:

```
harddisk:/usr/data/igmp/accounting.dat.RP_0_RSP0_CPU0.101225
```

The statistics file is stored on the route processor (RP) that is active. If a failover event occurs, then a new file is created on the new active RP, and no attempt is made to mirror the data between the active and the standby RP. Thus, the statistics files must be retrieved from both the active and standby RPs.

By default, the IGMP Accounting feature adds one file each day. To avoid exhausting disk space, you can specify in how many files, or for how many days, data should be retained, see Configuring IGMP Accounting, on page 253. Files older than the specified period are deleted, and the data is discarded from BNG. The maximum size of each file should be no more than 250 MB.

Configuring IGMP Accounting

Perform this task to configure the IGMP accounting.

SUMMARY STEPS

1. **configure**
2. **router igmp**
3. **accounting** [**max-history**] *days*
4. **commit**
5. **show igmp interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router igmp** **Example:** `RP/0/RSP0/CPU0:router(config)# router igmp`	Enter the router process for IGMP configuration mode.
Step 3	**accounting** [**max-history**] *days* **Example:** `RP/0/RSP0/CPU0:router(config-igmp-vrf1)# accounting max-history 50`	Configures the IGMP accounting. The max-history parameter is optional and specifies how many files are kept; this number is equivalent to the number of days in the history.
Step 4	**commit**	

	Command or Action	Purpose
Step 5	**show igmp interface** **Example:** `RP/0/RSP0/CPU0:router# show igmp interface`	(Optional) Displays IGMP interface information.

Configuring IGMP Accounting: An example

```
configure
router igmp
accounting max-history 45
!
!
end
```

DAPS Support

A Distributed Address Pool Service (DAPS) allows address pools to be shared between DHCP processes that run on a line card (LC) and the route processor (RP). The DHCP Server and PPPoE subscribers are clients to DAPS, and are known as the DAPS client. DAPS is used to return IP address to clients only when the RADIUS attributes contain the attribute "Pool Name". If the RADIUS attribute for a subscriber contains a fixed address, then the client does not contact DAPS for its IP address.

DAPS runs in two forms, as DAPS server on the RP, and as DAPS-Proxy on the LC. The RP has an in-build DAPS-Proxy module. This model ensures that all DAPS clients always talk to the DAPS-Proxy. The DAPS-Proxy instances talk to the central DAPS-Server on the RP for address assignments and other requests. DAPS-Proxy runs on all the LCs in the system. The DAPS-Proxy running on an LC can service multiple clients, from that LC; for example, PPP, DHCPv6, IPv6ND. DAPS serves multiple DAPS clients on two or more nodes. A separate DAPS-Proxy process runs on each node and connects locally to each DAPS Client.

DAPS supports dynamic IPv4 and IPv6 address allocation by pool name. For more information about configuring IPv4 DAPS, see Configuring IPv4 Distributed Address Pool Service, on page 254. To create a configuration pool for IPv6, see Creating a Configuration Pool Submode, on page 255.

You can configure various DAPS IPv6 parameters in the IPv6 configuration submode. You can configure the subnet number and mask for an IPv6 address pool, for more information, see Configuring the Subnet Number and Mask for an Address Pool, on page 256. You can specify parameters such as a range of IPv6 addresses. For more information, see Specifying a Range of IPv6 Addresses, on page 258. To specify a utilization threshold, see Specifying a Utilization Threshold, on page 258. To specify a set of prefixes or addresses inside a subnet, see Specifying a Set of Addresses or Prefixes Inside a Subnet, on page 261. You can also specify the length of a prefix. For more information, see Specifying the Length of the Prefix, on page 260.

Configuring IPv4 Distributed Address Pool Service

Perform this task to configure IPv4 distributed address pool service (DAPS).

SUMMARY STEPS

1. **configure**

2. **pool ipv4** *ipv4-pool-name*
3. **address-range** *first_address second_address*
4. **pool vrf** *vrf-name* **ipv4** *ipv4-pool-name*{**address-range** *address-range*}
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**pool ipv4** *ipv4-pool-name* **Example:** `RP/0/RSP0/CPU0:router(config)# pool ipv4 pool1`	Configures IPv4 pool name.
Step 3	**address-range** *first_address second_address* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv4)# address-range 1.1.1.1 9.8.9.8`	Configures the address range for allocation.
Step 4	**pool vrf** *vrf-name* **ipv4** *ipv4-pool-name*{**address-range** *address-range*} **Example:** `RP/0/RSP0/CPU0:router(config)# pool vrf vrf1 ipv4 pool1 address-range 1.1.1.1 9.8.9.8`	Configures IPv4 pool name.
Step 5	**commit**	

Configuring IPv4 Distributed Address Pool Service: An example

```
pool ipv4 pool1
address-range 1.1.1.1 9.8.9.8
pool vrf vrf1 ipv4 pool1 address-range 1.1.1.1 9.8.9.8
 !
!
end
```

Creating a Configuration Pool Submode

Perform this task to create and enable an IPv6 configuration pool submode for a default VRF and for a specific VRF.

SUMMARY STEPS

1. **configure**
2. **pool ipv6** *ipv6-pool-name*
3. **commit**

4. **configure**
5. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**pool ipv6** *ipv6-pool-name* **Example:** `RP/0/RSP0/CPU0:router(config)# pool ipv6 pool1`	Creates the IPv6 pool name for a default VRF and enters the pool IPv6 configuration submode.
Step 3	**commit**	
Step 4	**configure**	
Step 5	**pool vrf** *vrf_name* **ipv6** *ipv6-pool-name* **Example:** `RP/0/RSP0/CPU0:router(config)# pool vrf vrf1 ipv6 pool1`	Creates the IPv6 pool name for a specific VRF and enters the pool IPv6 configuration submode.
Step 6	**commit**	

Creating a Configuration Pool Submode: An example

```
configure
pool ipv6 pool1 (default vrf)
!
!
configure
pool vrf vrf1 ipv6 pool1 (for a specific vrf)
!
!
end
```

Configuring the Subnet Number and Mask for an Address Pool

Perform this task to create the subnet number and mask for an IPv6 address pool.

SUMMARY STEPS

1. **configure**
2. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
3. **prefix-length** *value*
4. **network** *subnet*
5. **utilization-mark high** *value* **low** *value*
6. **exclude** *low_ip_address high_ip_address*

7. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**pool vrf** *vrf_name* **ipv6** *ipv6-pool-name* **Example:** `RP/0/RSP0/CPU0:router(config)# pool vrf default ipv6 test`	Creates the IPv6 pool name for a specific VRF and enters the pool IPv6 configuration submode.
Step 3	**prefix-length** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-length 120`	Specifies the length of the prefix that is assigned to the clients. The value of the prefix length ranges from 1 to 128.
Step 4	**network** *subnet* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# network 1101:1::/114`	Specifies a set of addresses or prefixes inside a subnet. **Note** The **prefix-length** command must be mandatorily configured whenever the **network** command is used.
Step 5	**utilization-mark high** *value* **low** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# utilization-mark high 70 low 30`	Specifies a utilization threshold in the pool IPv6 submode. The high and low values are represented as percentages between 0 and 100.
Step 6	**exclude** *low_ip_address high_ip_address* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# exclude 1101:1::100 ::`	Specifies a range of IPv6 addresses or prefixes that DAPS must not assign to clients. The high and low values are represented as percentages between 0 and 100. **Note** Multiple exclude commands are allowed within a pool. To exclude a single address, <*high_ip_address*> can be omitted.
Step 7	**commit**	

Configuring the Subnet Number and Mask for an Address Pool: An example

```
configure
pool vrf default ipv6 test
prefix-length 120
network 1101:1::/114
utilization-mark high 70 low 30
exclude 1101:1::100 ::
!
```

```
!
end
```

Specifying a Range of IPv6 Addresses

Perform this task to specify a range of IPv6 addresses within a pool.

SUMMARY STEPS

1. **configure**
2. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
3. **address-range** *low_ip_address high_ip_address*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**pool vrf** *vrf_name* **ipv6** *ipv6-pool-name* **Example:** `RP/0/RSP0/CPU0:router(config)# pool vrf vrf1 ipv6 addr_vrf`	Creates the IPv6 pool name for a specific VRF and enters the pool IPv6 configuration submode.
Step 3	**address-range** *low_ip_address high_ip_address* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# address-range 1234::2 1234::3e81`	Specifies the range of IPv6 addresses within a pool. Multiple address-ranges are allowed within a pool.
Step 4	**commit**	

Specifying a Range of IPv6 Addresses: An example

```
configure
pool vrf vrf1 ipv6 addr_vrf
address-range 1234::2 1234::3e81
!
!
end
```

Specifying a Utilization Threshold

Perform this task to specify a utilization threshold for a specific VRF in the pool IPv6 submode.

SUMMARY STEPS

1. **configure**

2. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
3. **prefix-length** *value*
4. **network** *subnet*
5. **utilization-mark high** *value* **low** *value*
6. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**pool vrf** *vrf_name* **ipv6** *ipv6-pool-name* **Example:** `RP/0/RSP0/CPU0:router(config)# pool vrf default ipv6 test`	Creates the IPv6 pool name for a specific VRF and enters the pool IPv6 configuration submode.
Step 3	**prefix-length** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-length 120`	Specifies the length of the prefix that is assigned to the clients. The value of the prefix length ranges from 1 to 128.
Step 4	**network** *subnet* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# network 1101:1::/114`	Specifies a set of addresses or prefixes inside a subnet. **Note** The **prefix-length** command should be mandatorily configured whenever the **network** command is used.
Step 5	**utilization-mark high** *value* **low** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# utilization-mark high 70 low 30`	Specifies a utilization threshold in the pool IPv6 submode. The high and low values are represented as percentages between 0 and 100.
Step 6	**commit**	

Specifying a Utilization Threshold: An example

```
configure
pool vrf default ipv6 test
prefix-length 120
network 1101:1::/114
utilization-mark high 70 low 30
!
!
end
```

Specifying the Length of the Prefix

Perform this task to specify the length of the prefix that is assigned to the clients.

SUMMARY STEPS

1. **configure**
2. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
3. **prefix-length** *value*
4. **prefix-range** *low_ipv6_prefix high_ipv6_prefix*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**pool vrf** *vrf_name* **ipv6** *ipv6-pool-name* **Example:** `RP/0/RSP0/CPU0:router(config)# pool vrf vrf1 ipv6 prefix_vrf`	Creates the IPv6 pool name for a specific VRF and enters the pool IPv6 configuration submode.
Step 3	**prefix-length** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-length 64`	Specifies the length of the prefix that is assigned to the clients. The value of the prefix length ranges from 1 to 128.
Step 4	**prefix-range** *low_ipv6_prefix high_ipv6_prefix* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-range 9fff:1:: 9fff:1:0:3e7f::`	Specifies a range of IPv6 address prefixes for a specific VRF in the pool IPv6 configuration mode. **Note** The **prefix-length** must be mandatorily configured whenever **prefix-range** is configured.
Step 5	**commit**	

Specifying the Length of the Prefix that is Assigned to the Clients: An example

```
configure
pool vrf vrf1 ipv6 prefix_vrf
prefix-length 64
prefix-range 9fff:1:: 9fff:1:0:3e7f::
!
!
end
```

Specifying a Set of Addresses or Prefixes Inside a Subnet

Perform this task to specify a set of addresses or prefixes inside a subnet in the pool IPv6 configuration submode.

SUMMARY STEPS

1. **configure**
2. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
3. **prefix-length** *value*
4. **network** *subnet*
5. **utilization-mark high** *value* **low** *value*
6. **exclude** *low_ip_address high_ip_address*
7. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**pool vrf** *vrf_name* **ipv6** *ipv6-pool-name* **Example:** `RP/0/RSP0/CPU0:router(config)# pool vrf default ipv6 test`	Creates the IPv6 pool name for a specific VRF and enters the pool IPv6 configuration submode.
Step 3	**prefix-length** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# prefix-length 120`	Specifies the length of the prefix that is assigned to the clients. The value of the prefix length ranges from 1 to 128.
Step 4	**network** *subnet* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# network 1101:1::/114`	Specifies a set of addresses or prefixes inside a subnet. **Note** The **prefix-length** command should be mandatorily configured whenever the **network** command is used.
Step 5	**utilization-mark high** *value* **low** *value* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# utilization-mark high 70 low 30`	Specifies a utilization threshold in the pool IPv6 submode. The high and low values are represented as percentages between 0 and 100.
Step 6	**exclude** *low_ip_address high_ip_address* **Example:** `RP/0/RSP0/CPU0:router(config-pool-ipv6)# exclude 1101:1::100 ::`	Specifies a range of IPv6 addresses or prefixes that DAPS must not assign to clients. The high and low values are represented as percentages between 0 and 100.

	Command or Action	Purpose
		Note Multiple exclude commands are allowed within a pool. To exclude a single address, <high_ip_address> can be omitted.
Step 7	**commit**	

Specifying a Set of Addresses or Prefixes Inside a Subnet: An example

```
configure
pool vrf default ipv6 test
prefix-length 120
network 1101:1::/114
utilization-mark high 70 low 30
exclude 1101:1::100 ::
!
!
end
```

HTTP Redirect Using PBR

The HTTP Redirect (HTTPR) feature is used to redirect subscriber traffic to a destination other than the one to which it was originally destined. The HTTPR feature is implemented using Policy Based Routing (PBR) that makes packet forwarding decisions based on the policy configuration, instead of routing protocols. The HTTPR feature is implemented by sending an HTTP redirect response, which contains the redirect URL, back to the HTTP client that originally sent the request. Thereafter, the HTTP client sends requests to the redirected URL. HTTPR is supported for both IPv4 and IPv6 subscribers.

The most common use of HTTPR feature is for initial logon. In some cases, it is not possible to uniquely identify a subscriber and authorize them. This happens when the subscriber is using a shared network access medium to connect to the network. In such cases, the subscriber is allowed to access the network but restricted to what is known as an "open-garden". An open-garden is a collection of network resources that subscribers can access as long as they have physical access to the network. Subscribers do not have to provide authentication information before accessing the web sites in an open-garden.

When subscribers try to access resources outside the open-garden (which is called the "walled-garden"), they are redirected to a web logon portal. The walled-garden refers to a collection of web sites or networks that subscribers can access after providing minimal authentication information. The web logon portal requires the subscriber to login using a username and password. Thereafter, the web logon portal sends an account-logon CoA to BNG with user credentials. On successful authentication of these credentials, BNG disables the redirect and applies the correct subscriber policies for direct network access. Other uses of HTTPR include periodic redirection to a web portal for advertising reasons, redirection to a billing server, and so on.

The PBR function is configured in its own dynamic template. If the dynamic template contains other functions too, then the PBR policy that redirects packets must be deactivated using a CoA.

BNG maintains HTTP redirect statistics counters that track the number of packets that are being either redirected or dropped. The HTTP protocol uses some status codes to implement HTTPR. Currently, the redirect codes 302 (for HTTP version 1.0) and 307 (for HTTP version 1.1) are supported on BNG.

Note

- HTTP redirect applies only to HTTP packets. As a result, other services such as SMTP, FTP are not affected by this feature. Nevertheless, if these other services are part of the redirect classification rules, then the packets are dropped and not forwarded.
- HTTPS is not supported.
- Destination URL-based classification is not supported.
- HTTP redirect is supported only on subscriber interfaces.

The process of configuring HTTPR involves these stages:

- Creating access lists that define the redirected and open-garden permissions. See, Identifying HTTP Destinations for Redirection, on page 263.
- Creating the class-maps that uses the access list to classify the traffic as redirected, or permitted to access open-garden. See, Configuring Class Maps for HTTP Redirection, on page 266.
- Creating the policy-map to define the action to be performed on the traffic classified using class-maps. See, Configuring Policy Map for HTTP Redirect, on page 268.
- Creating the dynamic template to apply the service policy. See Configuring Dynamic Template for Applying HTTPR Policy, on page 270.

To configure a web logon that specifies a time limit to perform the authentication, see Configuring Web Logon, on page 271.

Identifying HTTP Destinations for Redirection

Perform this task to define access lists that identify http destinations that require redirection or are part of an open garden:

SUMMARY STEPS

1. **configure**
2. {**ipv4** | **ipv6**}**access-list** *redirect_acl_name*
3. Do one of the following:
    - [*sequence-number*]{**permit** | **deny**} *source* *source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
    - [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [*routing*] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
4. Repeat Step 3 as necessary, adding statements by sequence number. Use the **no** *sequence-number* command to delete an entry.
5. {**ipv4** | **ipv6**}**access-list** *open_garden_acl*
6. Do one of the following:

- [*sequence-number*]{**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
- [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [*routing*] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]

7. Repeat Step 6 as necessary, adding statements by sequence number. Use the **no** *sequence-number* command to delete an entry.
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	{**ipv4** | **ipv6**}**access-list** *redirect_acl_name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# ipv4 access-lists redirect_acl`
	or
	`RP/0/RSP0/CPU0:router(config)# ipv6 access-lists redirect_acl`
		Enters either IPv4 or IPv6 access list configuration mode and configures the named access list.

Step 3	Do one of the following:
	• [*sequence-number*]{**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
	• [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [*routing*] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
	Example:
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit 172.16.0.0 0.0.255.255`
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 deny 192.168.34.0 0.0.0.255`
	or
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit`
		Specifies one or more conditions allowed or denied in IPv4 or IPv6 access list redirect_acl.
		• The optional **log** keyword causes an information logging message about the packet that matches the entry to be sent to the console.
		• The optional **log-input** keyword provides the same function as the **log** keyword, except that the logging message also includes the input interface.
		or
		Specifies one or more conditions allowed or denied in IPv6 access list redirect_acl.
		• Refer to the **deny** (IPv6) and **permit** (IPv6) commands for more information on filtering IPv6 traffic based on based on IPv6 option headers and optional, upper-layer protocol type information.
		Note Every IPv6 access list has an implicit **deny ipv6 any any** statement as its last match condition. An IPv6 access list must contain at least one entry for the implicit **deny ipv6 any any** statement to take effect.

	Command or Action	Purpose
	`icmp any any` `RP/0/RSP0/CPU0:router(config-ipv6-acl)# 30 deny tcp any any gt 5000`	
Step 4	Repeat Step 3 as necessary, adding statements by sequence number. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise an access list.

Step 5	{**ipv4** | **ipv6**}**access-list** *open_garden_acl*
	Example:
	`RP/0/RSP0/CPU0:router(config)# ipv4 access-lists open_garden_acl`
	or
	`RP/0/RSP0/CPU0:router(config)# ipv6 access-lists open_garden_acl`
		Enters either IPv4 or IPv6 access list configuration mode and configures the named access list for open garden.

Step 6	Do one of the following:
	• [*sequence-number*]{**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
	• [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [*routing*] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
	Example:
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit 172.16.0.0 0.0.255.255`
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 deny 192.168.34.0 0.0.0.255`
	or
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit icmp any any`
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 30 deny tcp any any gt 5000`
		Specifies one or more conditions allowed or denied in IPv4 access list open_garden_acl.
		• The optional **log** keyword causes an information logging message about the packet that matches the entry to be sent to the console.
		• The optional **log-input** keyword provides the same function as the **log** keyword, except that the logging message also includes the input interface.
		or
		Specifies one or more conditions allowed or denied in IPv6 access list open_garden_acl.
		• Refer to the **deny** (IPv6) and **permit** (IPv6) commands for more information on filtering IPv6 traffic based on based on IPv6 option headers and optional, upper-layer protocol type information.
		Note Every IPv6 access list has an implicit **deny ipv6 any any** statement as its last match condition. An IPv6 access list must contain at least one entry for the implicit **deny ipv6 any any** statement to take effect.

Step 7	Repeat Step 6 as necessary, adding statements by sequence number. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise an access list.
Step 8	**commit**	

Identifying HTTP Destinations for Redirection: An example

```
configure
 ipv4 access-list <redirect-acl>
   10 permit tcp any any syn eq www
   20 permit tcp any any ack eq www
   30 permit tcp any any eq www
  ipv4 access-group <allow-acl>
   10 permit tcp any 10.1.1.0 0.0.0.255 eq www
   20 permit tcp any 20.1.1.0 0.0.0.255 eq www
   30 permit tcp any 30.1.1.0 0.0.0.255 eq www
   40 permit udp any any eq domain
 !
!
!
end
```

```
configure
 ipv6 access-list <redirect-acl>
   10 permit tcp any any syn eq www
   20 permit tcp any any ack eq www
   30 permit tcp any any eq www
  ipv6 access-group <allow-acl>
   10 permit tcp any 10.1.1.0 0.0.0.255 eq www
   20 permit tcp any 20.1.1.0 0.0.0.255 eq www
   30 permit tcp any 30.1.1.0 0.0.0.255 eq www
   40 permit udp any any eq domain
 !
!
!
end
```

Configuring Class Maps for HTTP Redirection

Perform this task to configure the class maps for HTTP redirection. It makes use of previously defined ACLs.

Before you begin

The configuration steps mentioned in Identifying HTTP Destinations for Redirection, on page 263 has to be completed before performing the configuration of the HTTPR class maps.

SUMMARY STEPS

1. **configure**
2. **class-map type traffic match-all** *open-garden-class_name*
3. **match** [**not**] **access-group**{**ipv4** | **ipv6**} *open_garden_acl*
4. **end-class-map**
5. **class-map type traffic match-all** *http_redirect-class_name*
6. **match** [**not**] **access-group** {**ipv4** | **ipv6**} *redirect_acl*
7. **end-class-map**
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**class-map type traffic match-all** *open-garden-class_name* **Example:** `RP/0/RSP0/CPU0:router(config)# class-map type traffic match-all CL1`	Defines a traffic class and the associated rules that match packets to the class for a open garden class.

Step 3	**match** [**not**] **access-group**{**ipv4** | **ipv6**} *open_garden_acl*
	Example:
	`RP/0/RSP0/CPU0:router(config-cmap)# match not access-group ipv4 open_garden_acl`
	or
	`RP/0/RSP0/CPU0:router(config-cmap)# match not access-group ipv6 open_garden_acl`
		Identifies a specified access control list (ACL) number as the match criteria for a class map.
		Note The redirect acl name provided in this step is the one configured in the configuration step mentioned in the prerequisites.

Step 4	**end-class-map** **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	Ends the configuration of match criteria for the class and exits the class map configuration mode.
Step 5	**class-map type traffic match-all** *http_redirect-class_name* **Example:** `RP/0/RSP0/CPU0:router(config)# class-map type traffic match-all RCL1`	Defines a traffic class and the associated rules that match packets to the class for a open garden class.

Step 6	**match** [**not**] **access-group** {**ipv4** | **ipv6**} *redirect_acl*
	Example:
	`RP/0/RSP0/CPU0:router(config-cmap)# match not access-group ipv4 redirect-acl`
	or
	`RP/0/RSP0/CPU0:router(config-cmap)# match not access-group ipv6 redirect-acl`
		Identifies a specified access control list (ACL) number as the match criteria for a class map.
		Note The redirect acl name provided in this step is the one configured in the configuration step mentioned in the prerequisites.

Step 7	**end-class-map** **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	Ends the configuration of match criteria for the class and exits the class map configuration mode.
Step 8	**commit**	

Configuring Class Maps for HTTP Redirection: An example

```
configure
class-map type traffic [match-any | match-all] <open-garden-class>
match [not] access-group ipv4 allow-acl
end-class-map

class-map type traffic [match-any | match-all] <http-redirect-class>
match [not] access-group ipv4 redirect-acl
end-class-map
  !
 !
!
end


configure
class-map type traffic [match-any | match-all] <open-garden-class>
match [not] access-group ipv6 allow-acl
end-class-map

class-map type traffic [match-any | match-all] <http-redirect-class>
match [not] access-group ipv6 redirect-acl
end-class-map
  !
 !
!
end
```

Configuring Policy Map for HTTP Redirect

Perform this task to configure policy maps for http redirect.

Before you begin

The configuration steps mentioned in Identifying HTTP Destinations for Redirection, on page 263 and Configuring Class Maps for HTTP Redirection, on page 266 have to be completed before performing the configuration of the policy-map for HTTPR.

SUMMARY STEPS

1. **configure**
2. **policy-map type pbr** *http-redirect_policy_name*
3. **class type traffic** *open_garden_class_name*
4. **transmit**
5. **class type traffic** *http_redirect-class_name*
6. **http-redirect** *redirect_url*
7. **class class-default**
8. **drop**
9. **end-policy-map**
10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**policy-map type pbr** *http-redirect_policy_name* **Example:** `RP/0/RSP0/CPU0:router(config)# policy-map type pbr RPL1`	Creates a policy map of type policy-based routing that can be attached to one or more interfaces to specify a service policy.
Step 3	**class type traffic** *open_garden_class_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic CL1`	Specifies the name of the class whose policy you want to create or change. **Note** The open garden acl name provided in this step is the one configured in the configuration step mentioned in the prerequisites.
Step 4	**transmit** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# transmit`	Forwards the packet to the original destination.
Step 5	**class type traffic** *http_redirect-class_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic RCL1`	Specifies the name of the class whose policy you want to create or change. **Note** The open garden acl name provided in this step is the one configured in the configuration step mentioned in the prerequisites.
Step 6	**http-redirect** *redirect_url* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# http-redirect redirect_url`	Specifies the URL to which the HTTP requests should be redirected.
Step 7	**class class-default** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# class class-default`	Configures default classes that cannot be used with user-defined classes.
Step 8	**drop** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# drop`	Drops the packet.
Step 9	**end-policy-map** **Example:**	Ends the configuration of a policy map and exits the policy map configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-cmap)# end-policy-map`	
Step 10	**commit**	

Configuring Policy Map for HTTP Redirect: An example

```
configure
policy-map type pbr <http-redirect-policy>
class type traffic <open-garden-class>
transmit
   !
class type traffic <http-redirect-class>
http-redirect <redirect-url>
!
class class-default
drop
  !
end-policy-map
  !
 !
!
end
```

Configuring Dynamic Template for Applying HTTPR Policy

Perform this task to configure dynamic template for applying the HTTPR policy to subscriber sessions.

Before you begin

The configuration steps mentioned in Configuring Policy Map for HTTP Redirect, on page 268 have to be completed before defining the dynamic template that uses a previously defined policy-map.

Note Ensure that the Dynamic template contains only the Policy Based Routing policy, so it can be easily deactivated after web login.

SUMMARY STEPS

1. **configure**
2. **dynamic-template type ipsubscriber** *redirect_template_name*
3. **service-policy type pbr** *http-redirect-policy*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**dynamic-template type ipsubscriber** *redirect_template_name* **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template type ipsubscriber RDL1`	Creates a dynamic template of type "ipsubscriber".
Step 3	**service-policy type pbr** *http-redirect-policy* **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# service-policy type pbr RPL1`	Attaches the service policy as a pbr type within a policy map created in the earlier configuration. **Note** The http redirect policy name provided in this step is the one configured in the configuration step mentioned in the prerequisites.
Step 4	**commit**	

Configuring Dynamic Template for Applying HTTPR Policy: An example

```
configure
dynamic-template type ip <redirect-template>
service-policy type pbr <http-redirect-policy>
  !
  !
!
end
```

Configuring Web Logon

Perform this task to configure Web Logon. As an example, a timer defines the maximum time permitted for authentication.

SUMMARY STEPS

1. **configure**
2. **class-map type control subscriber match-all** *classmap_name*
3. **match timer** *name*
4. **match authen-status authenticated**
5. **policy-map type control subscriber** *policymap_name*
6. **event session-start match-all**
7. **class type control subscriber** *class_name* **do-until-failure**
8. *sequence_number* **activate dynamic-template** *dt_name*
9. *sequence_number* **activate dynamic-template** *dt_name*
10. *sequence_number* **set-timer** *timer_name value*
11. **event account-logon match-all**
12. **class type control subscriber** *class_name* **do-until-failure**
13. *sequence_number* **authenticate aaa list default**
14. *sequence_number* **deactivate dynamic-template** *dt_name*

15. *sequence_number* **stop-timer** *timer_name*
16. **event time-expiry match-all**
17. **class type control subscriber** *class_name* **do-all**
18. *sequence_number* **disconnect**
19. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**class-map type control subscriber match-all** *classmap_name* **Example:** `RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-all IP_UNAUTH_COND`	Configures a subscriber control class-map with the match-all match criteria.
Step 3	**match timer** *name* **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# match timer AUTH_TIMER`	Configures a match criteria for the class along with timer details.
Step 4	**match authen-status authenticated** **Example:** `RP/0/RSP0/CPU0:router(config-cmap)# match timer AUTH_TIMER`	Configures a match criteria for the class along with authentication status details.
Step 5	**policy-map type control subscriber** *policymap_name* **Example:** `RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-all RULE_IP_WEBSESSION`	Configures a subscriber control policy-map.
Step 6	**event session-start match-all** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-all`	Configures the session start policy event that runs all the matched classes.
Step 7	**class type control subscriber** *class_name* **do-until-failure** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber class-default do-until-failure`	Configures the class to which the subscriber is to be matched. When there is a match, execute all actions that follow until a failure is encountered.
Step 8	*sequence_number* **activate dynamic-template** *dt_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 activate dynamic-template DEFAULT_IP_SERVICE`	Activates the dynamic-template defined locally on the CLI with the specified dynamic template name.

	Command or Action	Purpose
Step 9	*sequence_number* **activate dynamic-template** *dt_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 activate dynamic-template HTTP_REDIRECT`	Activates the dynamic-template defined locally on the CLI with the specified dynamic template name.
Step 10	*sequence_number* **set-timer** *timer_name value* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 set-timer AUTH_TIMER 4567`	Sets a timer to run a rule on its expiry. The timer value, specified in minutes, ranges from 0 to 4294967295.
Step 11	**event account-logon match-all** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-all`	Configures the account logon policy event that runs all matched classes.
Step 12	**class type control subscriber** *class_name* **do-until-failure** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber class-default do-until-failure`	Configures the class to which the subscriber is to be matched. When there is a match, execute all actions that follow, until a failure is encountered.
Step 13	*sequence_number* **authenticate aaa list default** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 authenticate aaa list default`	Specifies and authenticates the default AAA method list.
Step 14	*sequence_number* **deactivate dynamic-template** *dt_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 deactivate dynamic-template HTTP_REDIRECT`	Disables the timer before it expires.
Step 15	*sequence_number* **stop-timer** *timer_name* **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 20 stop-timer AUTH_TIMER`	Disables the timer before it expires.
Step 16	**event time-expiry match-all** **Example:** `RP/0/RSP0/CPU0:router(config-pmap)# event session-start match-all`	Configures the timer expiry policy event that runs all the matched classes.
Step 17	**class type control subscriber** *class_name* **do-all** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-e)# class type control subscriber IP_UNAUTH_COND do-all`	Configures the class to which the subscriber has to be matched. When there is a match, execute all actions.

	Command or Action	Purpose
Step 18	*sequence_number* **disconnect** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 disconnect`	Disconnects the session.
Step 19	**commit**	

Configuring Web Logon: An example

This example illustrates an IP session that is HTTP-redirected to an authentication web-portal for credentials. On successful authentication, the timer is unset. Otherwise, the subscriber gets disconnected when the timer window expires:

```
class-map type control subscriber match-all IP_UNAUTH_COND
   match timer AUTH_TIMER
   match authen-status unauthenticated

policy-map type control subscriber RULE_IP_WEBSESSION
    event session-start match-all
        class type control subscriber class-default do-until-failure
            10 activate dynamic-template DEFAULT_IP_SERVICE
            20 activate dynamic-template HTTP_REDIRECT
            30 set-timer AUTH_TIMER 5

    event account-logon match-all
        class type control subscriber class-default do-until-failure
            10 authenticate aaa list default
            15 deactivate dynamic-template HTTP_REDIRECT
            20 stop-timer AUTH_TIMER

    event timer-expiry match-all
        class type control subscriber IP_UNAUTH_COND do-all
            10 disconnect
```

Idle Timeout for IPoE and PPPoE Sessions

The Idle Timeout feature for IPoE and PPPoE sessions allows users to configure a maximum period of time that the subscriber sessions may remain idle. The subscriber sessions are terminated when this timeout period expires. The BNG monitors both the ingress and egress traffic for the determination of the idle time for the subscriber sessions. Control packets are not considered while determining session inactivity.

You can configure a threshold rate, and if packets sent or received by BNG in that interval is less than this threshold rate, then that particular session is considered idle. The threshold option allows you to consider low traffic rates as being idle and to exclude DHCP lease renewal packets from the statistics used for idle time determination. For instance, if you want to discount the DHCP short lease of 5 minutes, then you must configure the threshold as 5 packets per minute.

The dynamic template configuration of idle timeout is extended to also support **type ppp** templates. If idle timeout is enabled and if **monitor** action is not specified under the idle timeout event for a subscriber policy, then, by default, the sessions are disconnected. You can prevent the sessions from getting disconnected, by setting, for that particular subscriber policy, the policy action under the idle timeout event as **monitor**.

These Cisco VSAs are used to configure or update the idle timeout threshold and traffic direction from the RADIUS server:

```
idlethreshold = <mins/pkt>
idle-timeout-direction = <inbound | outbound | both>
```

For details on configuring idle timeout, see Creating Dynamic Template for IPv4 or IPv6 Subscriber Session, on page 77.

For details on configuring a policy-map with the idle-timeout event, see Configuring a Policy-Map, on page 65.

Routing Support on Subscriber Sessions

Routing support on subscriber sessions allows dynamic routes to be added on an individual subscriber basis for IPoE sessions. This allows to forward traffic from the default Virtual Routing and Forwarding (VRF) towards the subscriber, or to access the routes behind the subscriber. As opposed to static routes, dynamic routes must be added and removed when subscribers are created and deleted. Dynamic routes can belong to a VRF other than that of the subscriber and they are supported for IPv4 subscribers only.

Dynamic routes that are to be added for each subscriber are configured as part of the RADIUS profile of the subscriber. The subscriber sessions are not disconnected even if the dynamic route insertion fails. Instead, the route addition is re-tried at regular intervals.

The format of the Cisco:Avpair used for configuring the dynamic routes is:

```
Cisco:Avpair = "Framed-Route={vrf} [<destination_vrf>] {<prefix>} {<mask>} {vrf}
[<next_hop_vrf>] {<next_hop_ip_address>} [<admin_distance>] [tag <tag_value>]"
```

For example :

```
Cisco:Avpair = "Framed-Route=vrf vrfv1 10.121.1.254 255.255.255.255 vrf vrfv2 10.121.1.254
 30 tag 12"
```

In this example, the route for 10.121.1.254/32 is added to the vrfv1 with a next-hop of 10.212.1.254 in vrfv2. The route has an admin distance of 30 and a route tag value of 12.

Benefits of Routing Support on Subscriber Sessions

These are some of the benefits of routing support on subscriber sessions:

- Multiple dynamic routes for each subscriber are supported.
- The user can specify the destination VRF name and next-hop VRF name for each route to be added, and both can be different from the VRF of the subscriber. If the destination VRF is not specified, the VRF of the subscriber is taken as the default. If the next-hop VRF is not specified, the same VRF as that of the destination prefix is taken as the default.
- The user can specify the admin distance and tag to be used for the dynamic route.
- Dynamic routes are added as subscriber routes, with a default admin distance of 3.
- Dynamic routes are always recursive routes.
- Dynamic routes are CoA attribute and therefore, they can be changed while the subscriber is connected to the BNG router.

Traffic Mirroring on Subscriber Session

BNG supports the Traffic Mirroring feature on subscriber session. Traffic mirroring, also known as Switched Port Analyzer (SPAN), enables a user to monitor Layer 2 network traffic passing in or out of a set of Ethernet interfaces. This allows the mirroring of packets that pass through a source interface to a specified destination interface. The destination interface may then be attached to a network analyzer for debugging.

Traffic Mirroring or Switched Port Analyzer (SPAN) has these two distinct sets of configurations:

- Global configuration to create monitor sessions - A session is configured by specifying a session type and a destination that can be a local interface or a pseudo-wire interface.
- Source interface attachment configuration - This specifies how an interface should be attached to a monitor session.

For BNG, the source interface attachment configuration to a monitor session is through the use of dynamic templates. The subscriber is attached to the monitor session only when the template is applied to the subscriber. The template is applied or removed using the **activate-service** or **deactivate-service** CoA command sent from the RADIUS server or using the **test radius coa [activate | deactivate]** command.

For more information on Traffic Mirroring feature, see *Configuring Traffic Mirroring on the Cisco ASR 9000 Series Router* chapter in the *Interface and Hardware Component Configuration Guide for Cisco ASR 9000 Series Routers*. For complete command reference of the SPAN commands, see the *Traffic Mirroring Commands on the Cisco ASR 9000 Series Router* chapter in the *Interface and Hardware Component Command Reference for Cisco ASR 9000 Series Routers*.

For configuring traffic mirroring on BNG subscriber session, see Enabling Traffic Mirroring on Subscriber Session, on page 276.

Note

- It is recommended that a dynamic template is dedicated to SPAN configuration, so that SPAN can be enabled or disabled on a subscriber without any adverse impact.
- Modifications to SPAN configuration under a dynamic template, including the removal of configuration, have an immediate effect on all the subscribers to which that template is currently applied.

Enabling Traffic Mirroring on Subscriber Session

Perform this task to enable traffic mirroring on BNG subscriber session. These steps describe how to configure a dynamic template that references the monitor session and to associate or dis-associate it with a specific subscriber to enable or disable SPAN.

Before you begin

Create monitor sessions in global configuration mode using **monitor-session** command. Refer, Traffic Mirroring on Subscriber Session, on page 276

SUMMARY STEPS

1. **configure**

2. **dynamic-template type** {**ipsubscriber** | **ppp** | **service**} *dynamic-template-name*
3. Configure **monitor-session**, with optional **direction**, **acl** and **mirror first** options
4. **commit**
5. **test radius coa** {**activate** | **deactivate**} **service** *name* **acct-ses-id** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dynamic-template type** {**ipsubscriber** | **ppp** | **service**} *dynamic-template-name* **Example:** `RP/0/RSP0/CPU0:router(config)# dynamic-template type ppp ppp_template`	Creates a dynamic-template of type **ppp**.

Step 3	Configure **monitor-session**, with optional **direction**, **acl** and **mirror first** options
	Example:
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# monitor-session mon1 direction rx-only`
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# acl`
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# mirror first 100`
		Configures a dynamic template that references the monitor session.
		Note This syntax of monitor-session command for dynamic templates is same as the syntax for regular interfaces.

Step 4	**commit**	
Step 5	**test radius coa** {**activate** | **deactivate**} **service** *name* **acct-ses-id** *name* **Example:** `RP/0/RSP0/CPU0:router# test radius coa activate acct-ses-id 0x00000001 service service1`	If **activate** keyword is used, this command enables SPAN by associating a dynamic template with a specific subscriber. If **deactivate** keyword is used, this command disables SPAN by dis-associating a dynamic template with a specific subscriber.

Enabling Traffic Mirroring on Subscriber Session: An example

```
//Global configuration to create monitor sessions
configure
monitor-session mon1
destination interface gigabitethernet0/0/0/1
ethernet-services access-list tm_filter
 10 deny 0000.1234.5678 0000.abcd.abcd any capture
 !
!

//Configuring a dynamic template that references the monitor session
configure
dynamic-template type ppp ppp_template
 monitor-session mon1 direction rx-only
 acl
```

```
 mirror first 100
!
!

//Associating a dynamic-template with a specific subscriber to enable SPAN
test radius coa activate acct-ses-id 0x00000001 service service1
```

Randomization of Interim Timeout of Sessions or Services

The randomization feature distributes the interim timeouts in a relatively uniform manner and prevents accumulation of timeouts for interim accounts of sessions or services. This prevents a cycle where all messages are sent at once (this occurs if a primary link was recently restored and many dial-up users were directed to the same BNG at once). This is useful in scenarios such as churn scenarios of session bring up (that is, a small spurt with very high session bring up rate), subscriber redundancy group (SRG) slave to master switchover in BNG geo redundancy and so on.

For example, if a session is brought up at time 0, and it has an interim interval of 10 minutes (600 seconds), the first interim message is sent at time t1 = 600 seconds (this is without randomization enabled). With randomization enabled, a random number *x* which is less than 600 is selected and the first interim message is sent at that time, *x*. Use this command to specify the maximum variance allowed:

accounting interim variation

Sample configuration:

```
subscriber
 manager
  accounting interim variation 10
```

Additional References

These sections provide references related to implementing BNG subscriber features.

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 8

BNG Geo Redundancy

This chapter provides information about support of geographical redundancy through subscriber redundancy groups (SRGs) and session redundancy groups (SERGs).

Table 13: Feature History for Establishing Geo Redundancy

Release	Modification
Release 5.2.2	Introduced BNG geo redundancy.
Release 5.3.1	Geo redundancy support for PPPoE sessions was added.
Release 5.3.3	Peer route disable functionality was added.

This chapter covers these topics:

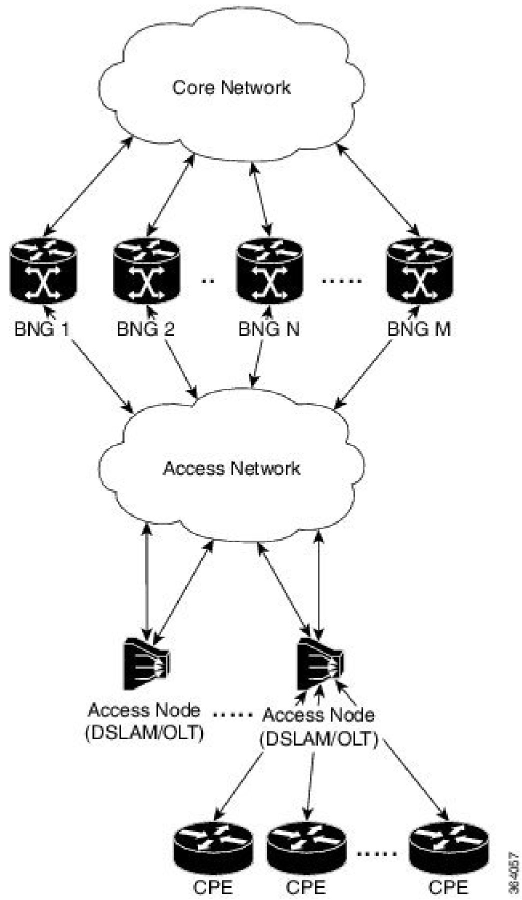
Geo Redundancy Overview

To provide redundancy for the subscriber sessions, BNG supports Geographical Redundancy across multiple BNGs, without having any L1 or L 2 connectivity between them. The BNG routers may be located in multiple geographical locations, and they have L3 connectivity over a shared core network through IP or MPLS routing.

Geo redundancy feature is supported for IPoE DHCP-triggered (IPv4, IPv6 and dual-stack) sessions and PPPoE (PTA and LAC) sessions.

This figure depicts a BNG geo redundancy deployment network model:

Figure 18: BNG Geo Redundancy Deployment Network Model



The redundancy pairing between BNG routers work by synchronizing the state from the master (active) to the slave (backup).

Geo redundancy works in conjunction with any of the access technologies. The CPEs are agnostic to redundancy; they see only one BNG or gateway. The access nodes are dual or multi-homed for redundancy using a variety of technologies based on the service provider network design and choices. Multi-chassis Link Aggregation (MC-LAG), dual-homed (Multiple Spanning Tree - Access Gateway or MST-AG), Ring (MST-AG or G.8032), xSTP and seamless MPLS (pseudowires) are a few such access networks.

Subscriber Redundancy Group (SRG)

Geo redundancy for subscribers is delivered by transferring the relevant session state from master BNG to slave BNG which can then help in failover (FO) or planned switchover (SO) of sessions from one BNG to another. Subscriber Redundancy Group (SRG) which is a set of access-interface (or a single access-interface) is introduced in BNG, and all subscribers in an SRG would FO or SO as a group.

The SRG has two modes of operation:

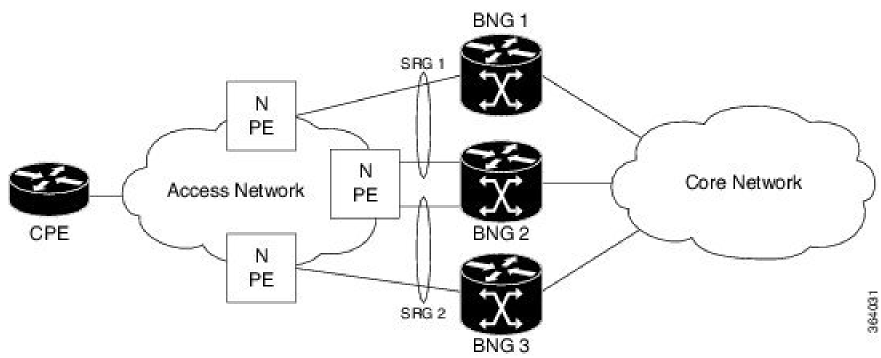
- Hot-standby
- Warm-standby

Currently BNG geo redundancy supports only the hot-standby slave mode. This is achieved by a 1:1 mirroring of subscriber session state from the master to the slave where the entire provisioning is done before the FO or SO. The sessions provisioned on slave is in sync with the set up on the master. Because the data plane is already set up for sub-second traffic impact, there is minimal action on switchover in the case of hot-standby mode and therefore, it is suitable for subscribers requiring high service level agreement (SLA). With appropriate capacity planning, the sessions can also be distributed across multiple BNGs to achieve an M: N model. The master-slave terminology is always in the context of a specific SRG; not for the BNG device as a whole.

Note Even after the Subscriber Redundancy Group (SRG) configuration is removed from the slave node, the CPE continues to receive ARP replies from both the master node and the slave node. This results in the network functioning in an uncertain manner. In order to avoid this uncertainty, shut down the access interface (that which corresponds to the slave node from which the configuration is removed) before removing the SRG configuration from the slave.

This figure depicts a typical BNG subscriber redundancy group (SRG):

Figure 19: BNG Subscriber Redundancy Group



SRG Virtual MAC

For seamless switchover between two BNGs, the L2-connected CPE devices must not detect change in gateway MAC and IPv4 or IPv6 addresses. The access technology like MC-LAG uses the same MAC address on both BNGs with active-standby roles, providing seamless switchover. Where MAC sharing is not provided by the access technology or protocol (like MST-AG, G.8032), the BNG SRG virtual MAC (vMAC) must be used. vMAC is configured as global MAC prefix or per SRG. This is integrated with BNG's dynamic master or slave role negotiation; additional protocols like VRRP or HSRP is not needed. vMAC (and its derived IPv6 link-local address) is used for control protocol exchanges (for example, ARP, ND, DHCP, PPPOE and so on) and data traffic for subscriber sessions or services only. It allows real port MAC to be used for Ethernet protocols (like E-OAM, xSTP, G.8032 and so on) that are leveraged by the SRG for doing failure detection, recovery and MAC Flush.

Session Distribution Across SRG

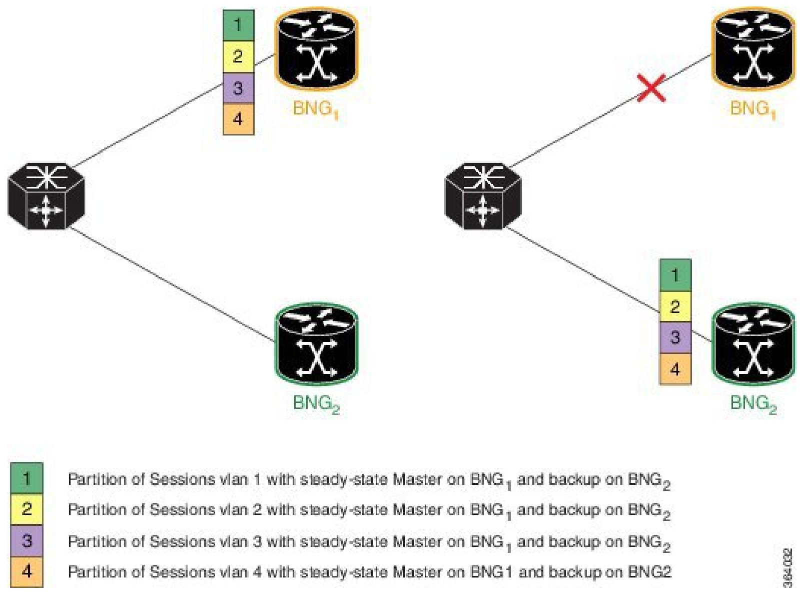
The session distribution across SRGs can be in either of these modes:

- Active-standby mode:

  In this mode, a dedicated backup BNG can be a slave for multiple SRGs from different active BNGs which are masters for those respective SRGs.

This figure shows an active-standby mode of session distribution across SRGs:

Figure 20: Active-standby Mode of Session Distribution



In figure a:

- Sessions are associated with partitions (VLAN 1, 2, 3 and 4) on BNG1, with each VLAN mapped to separate SRG configured as master role.
- BNG2 acts as backup for all VLANs.
- Each VLAN has 8K sessions terminated on it.

In figure b:

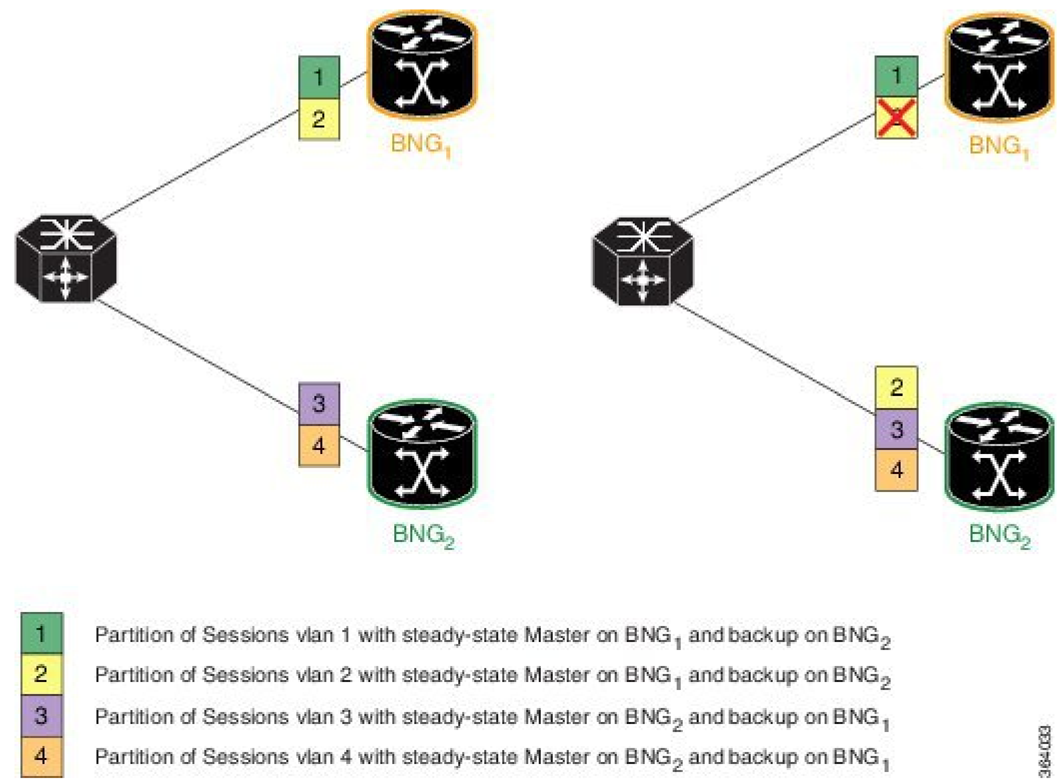
- An interface failure gets detected (using object-tracking of the access-interface) through MC-LAG.
- MC-LAG and SRG for each VLAN on BNG2 gets the master role.
- All 32K sessions are switched to BNG2.
- BNG2 sees a session termination count of 32K.

- Active-active mode:

  In this mode, a BNG can be master for one SRG and a slave for another SRG at the same time.

  This figure shows an active-active mode of session distribution across SRGs:

Figure 21: Active-active Mode of Session Distribution



In figure a:

- Sessions are associated with partitions (VLAN 1, 2) on BNG1, with each VLAN mapped to separate SRG configured as master role.
- Sessions are associated with partitions (VLAN 3, 4) on BNG2, with each VLAN mapped to separate SRG configured as master role.
- Each VLAN has 8K sessions terminated on it.
- Each BNG has 16K sessions terminated on it.

In figure b:

- The interface associated with VLAN 2 on BNG1 goes down.
- Sessions associated with partitions (VLAN 2) on BNG1 are switched to BNG2.
- BNG1 sees a session termination count of 8K and BNG2 sees a session termination count of 24K.

Benefits of BNG Geo Redundancy

Major benefits of BNG Geo Redundancy include:

- Supports various redundancy models such as 1:1 (active-active) and M:N, including M:1.
- Provides flexible redundancy pairing on access-link basis.

- Works with multiple access networks such as MC-LAG, dual-home and OLT rings.
- Supports various types of subscribers such as IPv4, IPv6 and dual-stack IPoE sessions.
- Works for RP (bundle and virtual access-links) based subscribers.
- Provides failure protection to access link failures, LC failures, RP failures and chassis failures.
- Performs automatic switchovers during dynamic failures or planned events such as maintenance, upgrades and transitions.
- Co-exists with other high availability (HA) or redundancy mechanisms.
- Does switchover of the impacted session group only; other session groups remain on the same BNG.
- Provides fast convergence and rapid setup of sessions, with minimal subscriber impact during switchover.
- Provides automatic routing convergence towards core and efficient address pool management.
- Provides seamless switchover for subscriber CPE without the need for any signaling.
- Integrates with RADIUS or policy and charging rule function (PCRF) systems.
- Provides minimal to zero incremental load on back end servers and PCRFs during normal operations and switchover.
- Does not impact session scale and call-per-second (CPS) during normal operation.

Supported Features in BNG Geo Redundancy

Supported Features in BNG Geo Redundancy

These access topologies are supported:

- MC-LAG topology (recommended only for IPv4 BNG sessions).
- Dual-home bundle interfaces with SRG vMAC using CFM or EFD fault detection and MST-AG for blocking.
- Ring bundle interfaces with SRG vMAC using CFM or EFD fault detection and MST-AG for blocking.
- Other access topologies and design variations may also be used for this feature.

These base geo redundancy features are supported:

- RP subscribers.
- Multiple SRG groups to different peer routers.
- Setting up peering statically through IPv4 or IPv6 TCP sessions.
- Hot-standby mode for slave (that is, subscribers provisioned in hardware on the slave as they are synchronized).
- Dynamic role negotiation between peers.
- Manual SRG switchover through command line interface (CLI).

- Dynamic failure detection using object tracking (link up-down, route and IPSLA tracking).
- Hold timer for dynamic switchover or switchback.
- Protocol bindings alone synchronized to slave; whereas AAA authorization for subscriber profile download performed by slave.
- Full BNG scale support (that is, half the scale number with redundancy).
- G.8032 (dual-home and ring) access technologies.

These DHCP features are supported:

- DHCPv6 IA-NA and IA-PD support for L2 connected sessions.
- DHCPv4 support for L2 connected sessions.
- DHCPv4 or DHCPv6 dual-stack support.
- DHCP proxy mode.
- Session initiation through DHCPv4 or DHCPv6 protocol.

Unsupported Features and Restrictions for BNG Geo Redundancy

This section lists the unsupported features and restrictions for BNG geo redundancy.

These are not supported in BNG geo redundancy:

- IPoE packet-triggered sessions.
- Routed (L3 connected) sessions
- Multicast
- Use of Neighbor Discovery (ND - SLAAC) feature for subscribers.
- SRG for ambiguous VLAN BNG sessions.
- SRG between Cisco IOS XR 64-bit BNG node and 32-bit BNG node.
- Both RP and LC subscribers do not support enabling fast switchover for subscriber framed-routes.

These are planned to be fully qualified only in future releases of Cisco IOS XR Software:

- Warm-standby slave mode.
- Line card (LC) based subscribers (that is, using physical port sub-interfaces).
- DHCP server mode.
- Pseudowire Headend (PWHE)

BNG Geo Redundancy Configuration Guidelines

While configuring BNG geo redundancy, certain guidelines must be followed in these areas:

- BNG Configuration Consistency

- Access-link Integration
- Core Routing Integration
- RADIUS-PCRF Integration

BNG Configuration Consistency

- Geo redundancy feature infrastructure synchronizes individual subscriber session state from master to slave. But, it does not synchronize the BNG related configurations (namely dynamic-template, DHCP profiles, policy-maps, access-interface configurations, external RADIUS or DHCP server and so on).
- For successful synchronization and setup of subscriber sessions between the two BNGs, it is mandatory that the relevant BNG configurations must be identical on the two routers and on the access-interfaces pairs in the SRG.
- While the access-interfaces or their types (or both) may vary between the paired BNGs, their outer-VLAN tag (that is, S-VLAN imposed by the access or aggregation devices) must be identical.
- Inconsistencies in base BNG or SRG configurations may result in synchronization failure and improper setup of sessions on the slave.

Access-link Integration

- You must use only those dual-homing techniques where one side is up or active, and the other side is down or standby. Both sides must not be up and forwarding traffic at the same time.
- You must use access-tracking mechanism under the SRG to ensure that its BNG role is always in synchronization with its access-link. Without this, the data or control traffic may get black-holed.
- The access-tracking object used by the SRG must be same as the one used in the routing configuration for conditional advertisement of the subscriber summary route(s) corresponding to that SRG's subscriber address or subnet pool(s).
- Including multiple access-links (which do not fail or switchover their roles) together into a single SRG may be challenging, unless mechanisms are implemented to ensure that all these links change state even when one of them fails.

Core Routing Integration

- Redistribution of individual subscriber routes into the routing protocol is not recommended because it slows convergence in failure or switchover events.
- Recommended design option is to conditionally advertise the summary static route for the subscriber address/subnet pool(s) of the SRG into the core routing protocol, through access-tracking.
- You can also advertise from both routers with different preferences and use various fast-reroute techniques.
- To avoid core routing changes in certain failure conditions, there are options to re-route the traffic from the slave to the master (for example, a tunnel or inter-chassis link) for transient or prolonged intervals.
- Routing convergence and its correlation with access failures or convergence is a key to overall end-to-end service impact for subscribers. Multiple options exist to achieve sub-second intervals.

RADIUS-PCRF Integration

The backend policy and charging rule function (PCRF) system must send the CoA message to both master and slave nodes. The message can be sent to the slave either at the same time as it is sent to master, or it can be sent after the slave takes over the master role and sends the Accounting START message.

From Cisco IOS XR Software Release R5.3.1 and later, the backend PCRF sytem need to send the CoA message only to the master node.

Session Sync

Once the session is up on the master node, the entire session information gets synced to the slave node. This includes dynamic synchronization of updates such as CoA or service logon. This is applicable from Cisco IOS XR Software Release R5.3.1 and later.

Setting up BNG Subscriber Redundancy Group

Guidelines in setting up SRG

Setting up SRG is subjected to these guidelines:

- The configurations and subscriber policies applied on the two routers (where the SRG access-interfaces are dual homing) must be identical to ensure seamless session mirroring and switchover.
  - SRG IDs (group IDs) must be same across BNGs.
  - Access-interface names or types need not be the same across routers.
  - Interface mapping-IDs must be same for the acces-interfaces across BNGs.
  - Server configurations (namely, RADIUS and DHCP configurations), IP pools, subscriber policies and templates must be identical across routers.
- The database of SRGs are scoped to a particular control plane instance (that is, at RP or LC node level). Therefore, you cannot form a single SRG with member links across LCs or with a mix of virtual interfaces (for example, bundles) and physical ports.

Setting up a BNG subscriber redundancy group (SRG) involves these steps:

- Enable BNG Geo-Redundancy:

```
subscriber redundancy
 source-interface loopback1
```

- Setup SRG and specify peer IPv4 or IPv6 address:

```
subscriber redundancy
 group 1
  peer 1.1.1.2
```

- Specify access-interfaces or VLANs, and mapping IDs:

```
subscriber redundancy
```

```
 group 1
  interface-list
    interface Bundle-Ether1.10 id 210
```

- Setup access object tracking for SRG and summary subscriber route:

```
track mclag-be1
 type line-protocol state
  interface bundle-ether1

subscriber redundancy
 group 1
  access-tracking mc-lag-be1

router static
 address-family ipv4 unicast
  200.0.0.0/16 Null0 track mc-lag-be1
```

Some optional configurations such as **preferred-role**, **slave-mode** and **hold-timer** also exist for SRG.

Geo Redundancy for PPPoE Sessions

BNG supports geo redundancy for PPPoE-PPP Termination and Aggregation (PPPoE-PTA) and PPPoE-L2TP Access Concentrator (PPPoE-LAC) sessions.

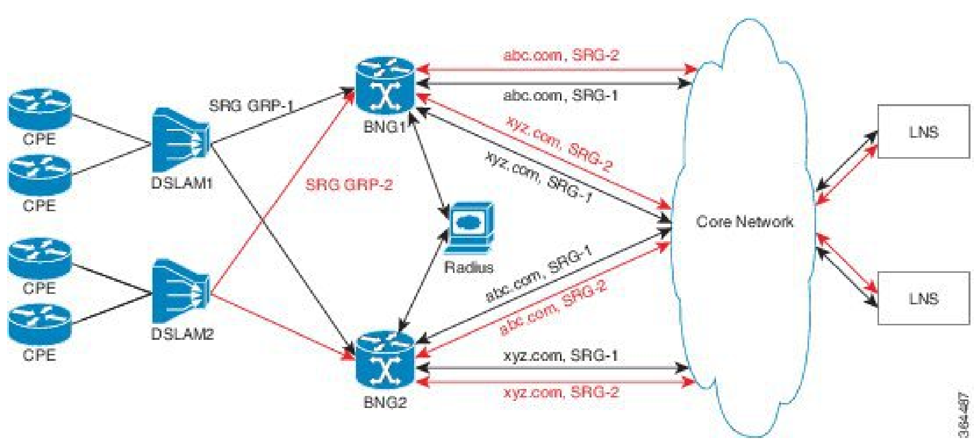
PPPoE-PTA Geo Redundancy

Geo redundancy behavior for the PPPoE-PTA sessions remains the same as for basic geo redundancy set up, except that the keepalives are disabled on the slave BNG node. The keepalives are sent only after the slave switches its role to master.

PPPoE-LAC Geo Redundancy

This figure shows a PPPoE-LAC Geo Redundancy set up with BNG

Figure 22: PPPoE-LAC Geo Redundancy Topology



For a PPPoE-LAC geo redundancy setup, the SRG is formed by grouping together the access-links on which LAC sessions are to arrive (co-exists with PTA). To enable SRG level redundancy switchover, tunnels for each SRG for each L2TP network server (LNS) must be setup. L2TP ensures that sessions belonging to different SRGs do not share the same tunnel even if they are going to the same LNS. The tunnel is set up on both master and slave nodes. By default, the tunnel is down on slave and it gets activated upon switchover. The BNG sync takes care of both tunnel and session-state sync from the master to the slave. The L2TP tunnel attributes and negotiated parameters are also synchronized through the BNG sync.

You must use this command in subscriber redundancy group configuration mode, to configure the source IP used for L2TP tunnel for subscribers coming under an SRG group:

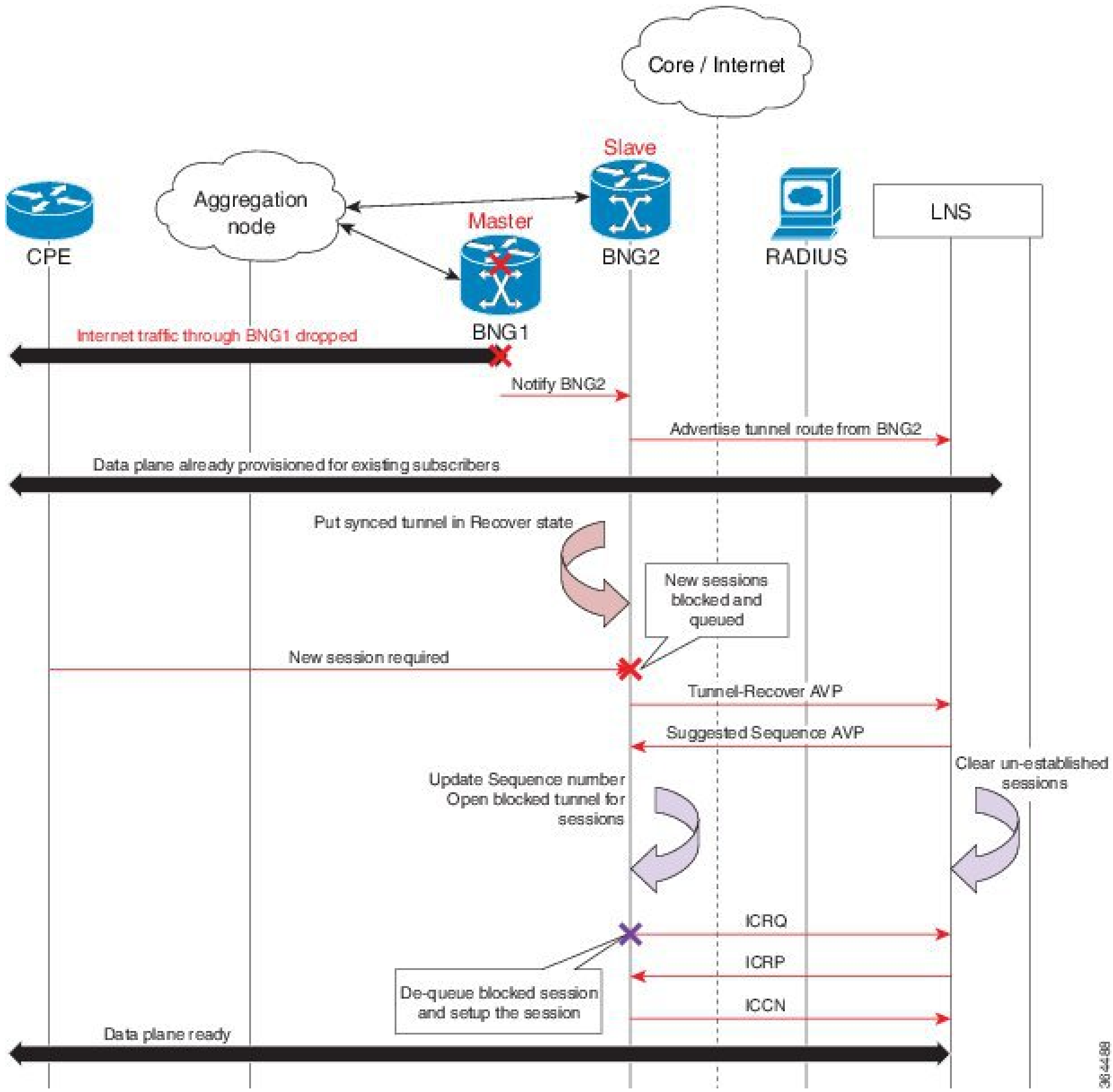
l2tp-source-ip *ipv4-address*

This ensures that there is a separate tunnel from each SRG group, in spite of having the same LNS.

PPPoE-LAC Session Switchover

This figure shows the call flow of PPPoE-LAC session switchover.

Figure 23: PPPoE-LAC Session Switchover



During switchover, the tunnel endpoint switches from the master (BNG1) to slave (BNG2) node as soon as the routing converges, and advertises the loopback address of slave (BNG2) to the LNS. The sessions and tunnels that are already provisioned on the data path on slave (BNG2) then seamlessly take over. The L2TP control plane on slave (BNG2) places the tunnel in re-sync state to recover the tunnel sequence number (Ns and Nr) during which only control messages are queued up for further processing. After the tunnel recovery, the LAC gets the sequence number from the LNS. The existing tunnels or sessions are not lost as the slave (BNG2) takes over. The signaling for the new session resumes and the queued requests also get processed. The unestablished sessions are then cleared off. For LNS, this switchover appears to be a convergence event where the tunnel has flapped.

Verification of Geo Redundancy for PPPoE Sessions

Listed below are some of the show commands that can be used to verify the Geo Redundancy configuration in BNG. For complete command reference, see the *Subscriber Commands*, *PPPoE Commands* and *PPPoE LAC-Specific Commands*, chapters in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*.

- **show subscriber redundancy group** *210*

```
Subscriber Redundancy Group ID: 210
  Description : <<not-configured>>

  Status                 : Enabled
  Init-Role              : Master
  Negotiated-Role        : Master                 Current-Role : Master

  Slave-mode             : Hot                       Hold Time : 15
  - - -
  - - -
Peer:
    11::2                                               Status : Established
    Role(Init/Neg/Cur): Slave/Slave/Slave
    Tracking Status    : Down
- - -
- - -
Switchover:
    Last Switchover    : 2014 Sep 12 07:12:11       Reason : Object Tracking Status
Change
- - -
- - -
Subscriber Session Statistics:
    Count              : 8000               Slave-Upd-Fail : 0
    Pending Update     : 0                  Pending Delete : 0
    Tunnel Count       : 0

Interface Count     : 1
    Bundle-Ether1.10                        Map-ID         : 210
```

- **show ppp interfaces**

```
Bundle-Ether2.1.pppoe16534 is up, line protocol is up
 SRG Role: Slave
  LCP: Open
     Keepalives enabled (60 sec, retry count 5)
     Local MRU: 1492 bytes
     Peer  MRU: 65531 bytes
  Authentication
     Of Peer: PAP (Completed as user1@domain.com)
     Of Us:   <None>
  IPCP: Open
     Local IPv4 address: 12.16.0.1
     Peer IPv4 address:  12.0.250.23
  IPv6CP: Initial
     Local IPv6 address: fe80::
     Peer IPv6 address:  fe80::
```

- **show pppoe interfaces**

```
Bundle-Ether2.1.pppoe16534 is Complete
```

```
  Session id: 16534
  Parent interface: Bundle-Ether2.1
  BBA-Group: BBA1
  Local MAC address: 0002.0003.0004
  Remote MAC address: 0000.6201.0103
  Outer VLAN ID: 10
  Tags:
    Service name: AGILENT
    Host-Uniq: 4 bytes, (000e0000)
  SRG-state: SRG-Standby
```

- **show vpdn**

```
RP/0/RSP0/CPU0:router# show vpdn session

SRG Role: Master
Subscriber label: 0x42, interface name: Bundle-Ether1.10.pppoe3
 user name: user1@lns2.com
 parent interface: Bundle-Ether1.10
 state: est last change: 00:01:01
 time to setup session:   0:2  (s:msec)
 conditional debug flags: 0
 L2TP data
  local end point: 11.1.1.1 remote end point: 19.9.9.2
  call serial number: 1970100002
  local tunnel id: 46813 remote tunnel id: 40849
  local session id: 36198 remote session id: 33437 remote port: 1701
  tunnel assigned id:
  tunnel client authentication id: LAC
  tunnel server authentication id: LNS
  tunnel authentication: disabled
  class attribute mask:
 Subscriber data
  NAS port id: 0/0/1/10
  NAS port type: Virtual PPPoE over VLAN
  physical channel id: 0
  Rx speed: 1000000000, Tx speed: 1000000000
 Configuration data
  table id: 0xe0000000, VRF id: 0x60000000, VPN id: 0:0
  VRF name: default
  dsl line info forwarding: disabled, l2tp busy timeout: 60
  TOS mode: default
```

Geo Redundancy Features

Peer Route Disable

Peer route disable is an enhancement in BNG geo redundancy whereby the user can disable the route on geo redundancy hot-standby peer. This disabling is so that the subscriber routes are not installed in the RIB even when the subscriber sessions are brought up on the standby peer. The subscriber routes are inserted into the RIB only when the BNG Geo-Redundancy state of peer changes to active. This ensures that only the master BNG, and not the slave BNG, routes the subscriber traffic in a scenario where access-interface is up on the standby peer. By disabling the routes, the hot-standby mode in BNG geo redundancy does not mandate the access-interface to be down on the standby peer any more.

To enable this feature, use the **peer route disable** command in subscriber redundancy group configuration mode.

Configuration Example

```
RP/0/RSP0/CPU0:router(config)# subscriber redundancy group 110
RP/0/RSP0/CPU0:router(config-subscr-red-group)# peer route-disable
```

Deployment Models for BNG Geo Redundancy

Multiple access networks are considered for BNG geo redundancy deployment scenarios. Some of the sample use cases are:

- Multi-chassis Link Aggregation (MC-LAG) - Two BNG boxes that are point-of-attachment (POA) devices, connected through MC-LAG either to a single Dual Homed Device (DHD) or to a DHD-pair using MC-LAG.
- Multiple Spanning Tree - Access Gateway (MST-AG):
  - Dual Homed Device using Bundle Interfaces - A single DHD with one bundle interface each to the two BNGs in active-active mode.
  - Ethernet Access Network-Ring - A physical ring (open or closed) that connects multiple OLTs (or L2 devices in general) to the two BNGs in active-active mode.

CHAPTER 9

DIAMETER Support in BNG

DIAMETER provides a base protocol that can be extended in order to provide authentication, authorization, and accounting (AAA) services to new access technologies. This chapter provides information about DIAMETER protocol and its support in BNG.

Table 14: Feature History for DIAMETER Support in BNG

Release	Modification
Release 5.3.0	This chapter was introduced for DIAMETER support feature in BNG.

This chapter covers these topics:

DIAMETER Overview

DIAMETER is a peer-to-peer protocol that is composed of a base protocol and a set of applications that allow it to extend its services to provide AAA services to new access technologies. The base protocol provides basic mechanisms for reliable transport, message delivery, and error handling and the base protocol must be used in conjunction with a DIAMETER application. Each application relies on the services of the base protocol to support a specific type of network access. Each application is defined by an application identifier and associated with commands. Each command is defined with mandatory Attribute Value Pairs (AVPs) and non-mandatory AVPs including vendor-specific AVPs.

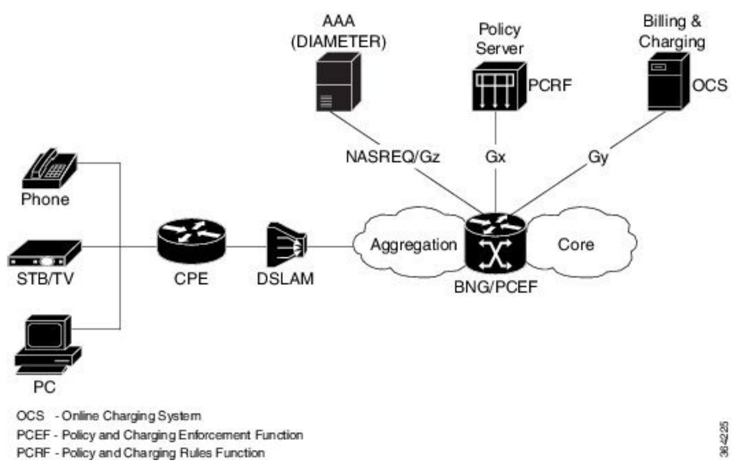
DIAMETER allows peers to exchange a variety of messages. The DIAMETER client generates DIAMETER messages to the DIAMETER server to perform the AAA actions for the user. This protocol also supports server-initiated messages, such as a request to abort service to a particular user.

DIAMETER Interface in BNG

BNG supports the DIAMETER base protocol, along with applications such as DIAMETER Credit Control Application (DCCA) and Network Access Server Requirements (NASREQ), which is used for policy control and charging, and real-time credit control of pre-paid users. BNG acts as NASREQ and DCCA client to perform AAA NAS related functionality, policy provisioning, quota request and usage reporting function. With this DIAMETER interface, BNG provides service-aware billing functionality and policy provisioning for post-paid and pre-paid users.

This figure shows the network of the DIAMETER interface in BNG:

Figure 24: DIAMETER Interface in BNG



Along with the DIAMETER base protocol, these DIAMETER applications are also supported in BNG:

- Diameter Credit Control Application (DCCA)
- Gx interface for Policy Control and Charging
- Gy interface for online charging
- Gz interface for offline charging

This table lists IANA-assigned application IDs for DIAMETER applications:

DIAMETER Application	DIAMETER Application ID
DIAMETER common message	0x00000000
DIAMETER NASREQ message	0x00000001
DIAMETER base accounting	0x00000003
DIAMETER DCCA application(Gy)	0x00000004

DIAMETER Application	DIAMETER Application ID
DIAMETER policy interface(Gx)	0x01000016 (16777224)

Features supported for BNG with DIAMETER

These base protocol features are supported in BNG with DIAMETER:

- TCP as the transport protocol for DIAMETER messages
- TLS support over TCP for secure communication
- IPv4 and IPv6 transport stack to the back end DIAMETER server

These base protocol features are not supported in BNG with DIAMETER:

- Communication with diameter peers that act as proxy, relay or a redirection agent
- Diameter peer discovery
- SCTP as the transport protocol for DIAMETER messages
- Internet Protocol Security (IPSec)

Supported DIAMETER Base Messages

BNG supports these DIAMETER base messages:

DIAMETER Base Messages	Abbreviation	Command Code	Description
Capabilities-Exchange-Request	CER	257	Sent from the client to the server to determine the capabilities of the server.
Capabilities-Exchange-Answer	CEA	257	Sent from the server to the client in response to a CER message.
Disconnect-Peer-Request	DPR	282	Sent to the peer to inform about the termination of the connection. The client or server may initiate the termination.
Disconnect-Peer-Answer	DPA	282	Sent as a response to a DPR message.
Device-Watchdog-Request	DWR	280	Sent from the client to the server to monitor the health of the connection. This happens if, for a while, there is no traffic between peers, after CER and CEA messages are exchanged.
Device-Watchdog-Answer	DWA	280	Sent as response to a DWR message.

For details of DIAMETER attributes and sample packets of DIAMETER messages, see Appendix E, DIAMETER Attributes, on page 393.

DIAMETER NASREQ Application

The NASREQ application is used for Authentication, Authorization and Accounting (AAA) in the Network Access Server (NAS) environment. For subscriber authentication or authorization, as part of the session creation, a DIAMETER AA-Request message is sent to the DIAMETER NASREQ server and the response may be an AA-Answer message. Subscriber accounting for sessions and services is done using AC-Request and AC-Answer messages of the NASREQ application. BNG supports the NASREQ application for network access related functionality; the admin access requests (such as Telnet, SSH, rlogin, and so on) must not be transported using the DIAMETER protocol. Because Extensible Authentication Protocol (EAP) authentication is not required in BNG, the support for DIAMETER EAP application is not considered.

If the user deploys a separate Offline Charging Server (OFCS) with the AAA method list configuration, the NASREQ application forwards the messages accordingly.

No new application-specific AVPs are sent for the NASREQ application, except DIAMETER-specific common set of AVPs and RADIUS prohibited AVPs for accounting.

This table lists the DIAMETER NAS messages supported by BNG:

DIAMETER NAS Messages	Abbreviation	Command Code	Description
AA-Request	AAR	265	Used to request authentication or authorization (or both) for a given NAS user. Admin user related AVPs are not applicable for BNG deployment with DIAMETER NASREQ application.
AA-Answer	AAA	265	Sent in response to the AAR message. If authorization was requested, a successful response includes the authorization AVPs appropriate for the service being provided. For backward compatibility and also based on the session type if it is IPoE or PPPoE, a few additional DIAMETER Cisco VSAs may also be present in this message.
Re-Auth-Request	RAR	258	Sent by a DIAMETER server when it initiates a re-authentication or re-authorization (or both) service for a particular session.
Re-Auth-Answer	RAA	258	Sent in response to the RAR message. The Result-Code AVP must be present in the RAA message and it indicates the disposition of the request. A successful RAA transaction must be followed by an AAR message.

DIAMETER NAS Messages	Abbreviation	Command Code	Description
Session-Termination-Request	STR	275	Sent by NAS to inform DIAMETER server that an authenticated or authorized (or both) session is being terminated. This is required only if NASREQ application is stateful.
Session-Termination-Answer	STA	275	Sent by DIAMETER server to acknowledge the session termination notification sent by NAS. The Result-Code AVP must be present in this STA message, and it may also contain an indication that an error occurred while the STR was being serviced. Upon sending or receiving the STA, the DIAMETER server must releases all resources for the session indicated by the Session-ID AVP.
Abort-Session-Request	ASR	274	Sent by DIAMETER server to NAS to stop the session identified by the Session-ID AVP. This is similar to RADIUS CoA Session-disconnect request or POD. In the case of stateless application, the DIAMETER session with the particular Session-ID does not exist on BNG. Therefore, instead of Session-ID, another BNG subscriber identity such as Acct-Session-ID, <Framed-IP-Address, VRF> may be sent as one of the AVPs.
Abort-Session-Answer	ASA	274	Sent in response to the ASR message. These are the possible result codes: • DIAMETER_SUCCESS - If the session identified by Session-ID was successfully terminated. • DIAMETER_UNKNOWN_SESSION_ID - If the session is not currently active. • DIAMETER_UNABLE_TO_COMPLY - If the access device does not stop the session for some reason.
Accounting-Request	ACR	271	Sent by a DIAMETER node that is acting as a client, in order to exchange accounting information with a peer. In addition to the standard AVPs, ACR messages must also include service-specific accounting AVPs.

DIAMETER NAS Messages	Abbreviation	Command Code	Description
Accounting-Answer	ACA	271	To acknowledge an ACR message. The ACA message contains the same Session-ID as the corresponding request.

DIAMETER Accounting

The session accounting and service accounting functionality provided by BNG, remain unchanged with the introduction of the DIAMETER interface. BNG uses accounting messages defined in the DIAMETER base protocol. The DIAMETER NASREQ application is used for regular AAA services over DIAMETER. The DIAMETER accounting message construction and transport is supported as part of this application.

The DIAMETER applications in BNG have the option of using either or both of these accounting application extension models:

- Split Accounting Service - The accounting message carries the Application-ID of the DIAMETER base accounting application (0x00000003). The respective diameter nodes advertise the DIAMETER base accounting Application ID during capabilities exchanges (CER and CEA).
- Coupled Accounting Service - The accounting message carries the Application-ID of the application that is using it (for example, NASREQ). The application itself processes the received accounting records or forwards them to an accounting server. The accounting application advertisement is not required during capabilities exchange, and the accounting messages are routed the same way as any of the other application messages. In the case of BNG, where an application does not define its own accounting service, the use of the split accounting model is preferred.

  The Gz interface between PCEF and OFCS use DIAMETER base accounting application for offline charging. Because BNG supports session based and service based accounting, the split accounting model in which the accounting Application-ID is inserted in all the accounting messages, is preferable.

BNG does not support persistence of accounting records when the DIAMETER server is down.

DIAMETER Accounting Messages

Accounting-Request (ACR) and Accounting-Answer (ACA) are the typical DIAMETER accounting NASREQ messages. The possible ACR types are:

1. EVENT_RECORD - sent if a session fails to start, along with the reason for the failure.
2. START_RECORD - sent if the first authentication or authorization transaction is successfully completed.
3. INTERIM_RECORD - sent if additional authentications or authorizations occur.
4. STOP_RECORD - sent upon termination of the session context.

DIAMETER Gx and Gy Applications

The Gx reference point (based on 3GPP TS 129 212 V11.10.0), that is located between Policy and Charging Rules Function (PCRF) and Policy and Charging Enforcement Function (PCEF), is used for provisioning and removal of policy and charging control (PCC) rules from the PCRF to the PCEF and for the transmission of traffic plane events from PCEF to PCRF. BNG acts as a PCEF in the current deployment. The PCRF acts as a DIAMETER server with respect to the DIAMETER protocol defined over the Gx interface. That is, it is the network element that handles PCC rule requests for a particular realm. The PCEF acts as the DIAMETER client. That is, it is the network element that requests PCC rules in the transport plane network resources. Currently BNG supports the Gx interface for PCC rules provisioning, but the usage monitoring feature on Gx interface (3GPP RLS9) is not supported.

The Gy reference point (based on 3GPP TS 132 299 V11.9.1), that is located between OCS and PCEF, is used for reporting and online charging.

The required AVPs for broadband deployment and for Cisco ASR 9000 Series Aggregation Services Router use cases are derived out of the Gx and Gy reference points.

Note When there is a DIAMETER process restart, all the ongoing or transient Gy sessions corresponding established Gx sessions are dropped for the sessions between a customer premise equipment (CPE) and the network resource.

Supported Gx Messages

This table lists the DIAMETER Gx messages supported by BNG:

DIAMETER Gx Messages	Abbreviation	Command Code	Description
Credit-Control-Request	CCR	272	Sent by the traffic plane function (TPF) to the charging rules function (CRF) in order to request charging rules for a bearer, and also to indicate the termination of the subscriber session.
Credit-Control-Answer	CCA	272	Sent by the PCRF to the PCEF in response to the CCR command. It is used to provision PCC rules and event triggers for the bearer or session, and to provide the selected bearer control mode for the IP connectivity access network (IP-CAN) session.
Re-Auth-Request	RAR	258	Sent by the PCRF to the PCEF in order to provision unsolicited PCC rules using the PUSH procedure.
Re-Auth-Answer	RAA	258	Sent by the PCEF to the PCRF in response to the RAR command.
Abort Session Request	ASR	274	Sent by any server to the access device providing session service, requesting it to stop the session identified by the Session-Id.
Abort Session Answer	ASA	274	Sent in response to the ASR. The Result-Code AVP that indicates the disposition of the request must be present.

Supported Gy Messages

BNG supports these DIAMETER Gy messages:

- CCR-Initial
- CCA-Initial
- CCR-Update message with tariff change units
- CCA-Update
- CCR-Final
- CCA-Final

DIAMETER DCCA Application

DCCA interface implementation is based on the RFC 4006. The 3GPP Gx and Gy applications use the DCCA framework and AVPs to provide the respective functions.

BNG supports these DCCA messages:

- Credit Control Request (CCR)
- Credit Control Answer (CCA)

Every single CCR must be responded with a separate CCA.

DCCA Session and Services

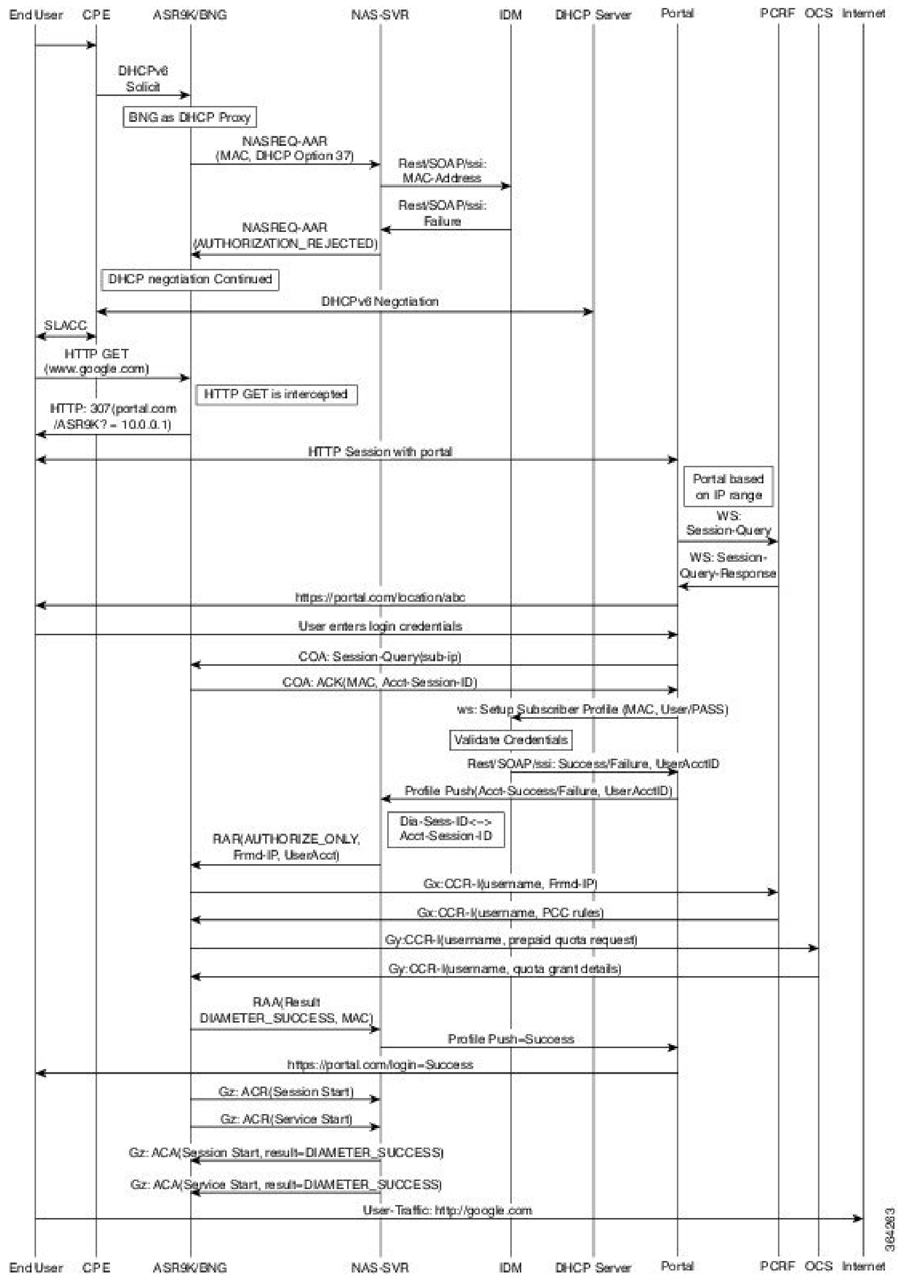
Each BNG subscriber session is associated with a DIAMETER CC-Session (Credit Control-Session) when Gx or Gy, or both applications, are enabled. Multiple services may be active in a BNG subscriber session. The quota management and usage reporting for each service is performed by using MSCC AVP in the CCR-CCA messages. The Service-Identifier and Rating-Group AVP inside the MSCC identifies the service of a subscriber session. Quota for a service is granted within one Granted-Service-Unit AVP (GSU). Quota usage reporting is done in one or more Used-Service-Unit (USU) AVP.

A CC-Session is uniquely identified by a Diameter Session-ID. The same format is used for the construction of Session-ID.

BNG DIAMETER Call Flow

This figure shows a call flow sequence of BNG DIAMETER, for DHCP-initiated IPoE sessions (this is based on one of the BNG DIAMETER use cases and the BNG call flow):

Figure 25: BNG DIAMETER Call Flow



Guidelines and Restrictions for DIAMETER Support in BNG

Guidelines for DIAMETER AVPs in BNG

These guidelines must be taken into consideration for the DIAMETER AVPs in BNG:

- Because BNG is deployed in wire-line scenario, Subscription-ID (443) AVP is not required. Instead, the subscriber identifier is carried using DIAMETER User-Name (1) AVP. If a provider likes to use the common subscriber identity, BNG can include Subscription-ID(443) Grouped AVP with the appropriate value for Subscription-ID-Type (450).
- To bring up a BNG session, a few Cisco VSAs are also needed as part of the subscriber authorization profile. Since the profile is provided by the PCRF, you must ensure the support of those DIAMETER Cisco AVPs.
- The network access details are sent from BNG in the request packet using the existing RADIUS equivalent of DIAMETER AVPs, such as NAS-Port-ID (87), NAS-Identifier (32) and NAS-IP-Address (4).
- The user must define the subscriber service on the BNG router as part of the dynamic template. The configurations on BNG router defines the service definitions that are part of a prepaid set. Hence, from the Gx interface perspective, only the Service-name is expected to come from PCRF. More than one service-name instance may come in CCA and RAR messages from PCRF. BNG receives these instances using Charging-Rule-Install (1001) 3GPP Grouped AVP, Charging-Rule-Name (1005) 3GPP AVP, Service-Identifier (439) IETF AVP and Rating-Group (432) 3GPP AVP, to be part of this grouped AVP to represent the one logical service construct.
- Currently BNG does not support service definition coming from PCRF. Therefore, the Charging-Rule-Definition(1003) 3GPP Grouped AVP, with containers to denote the flow-description, is not required.

Restrictions for DIAMETER in BNG

The DIAMETER support in BNG is subjected to these restrictions:

- BNG does not support Origin-State-Id AVP. Therefore, if this AVP is received from the DIAMETER server, it is ignored.
- The Session-Binding AVP is ignored by BNG router. BNG uses the value of Origin-Host AVP, received in the latest CCA message, for the Destination-Host AVP of the next request and the termination request as well.
- The use of In-band-Security-Id AVP, that is used to advertise the support of security portion of the application is not recommended in CER and CEA messages. Instead, discovery of a DIAMETER entity's security capabilities can be done through static configuration.

Configuring DIAMETER Peer in BNG

Perform this task to configure the DIAMETER connection on a BNG router.

The selection of DIAMETER server is mostly based on the AAA method list configuration. These are the various selection options:

- For regular AAA services (NASREQ), it is completely based on the AAA configuration on the router.
- For Gx, it can be based on the Gx realm selection.
- For prepaid, it is based on the charging profile associated with the subscriber session on BNG.

For details on configuring AAA for DIAMETER, see Configuring AAA for DIAMETER Peer in BNG, on page 310.

SUMMARY STEPS

1. **configure**
2. **diameter** {**gx** | **gy**}
3. **diameter peer** *peer name*
4. **transport security-type tls**
5. **transport tcp port** *port_num*
6. **destination host** *host_string*
7. **destination realm** *realm_string*
8. **address** [**ipv4** | **ipv6**] *ip_addr*
9. **ip vrf forwarding** *vrf_table_name*
10. **source-interface** *intf-type intf-name*
11. **peer-type server**
12. **root**
13. **diameter origin host** *host-name*
14. **diameter origin realm** *realm-string*
15. **diameter timer** [**connection** | **transaction** | **watchdog**] *timer-value*
16. **diameter vendor supported** [**cisco** | **etsi** | **threegpp** | **vodafone**]
17. **diameter tls trustpoint** *label*
18. **diameter** {**gx** | **gy**} [**retransmit** *retansmit-timer-val* | **tx-timer** *tx-timer-val*]
19. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**diameter** {**gx** | **gy**} **Example:** `RP/0/RSP0/CPU0:router(config)# diameter gx`	Configures Gx interface for policy control and charging. Similarly, configures the Gy interface for online (prepaid) charging.
Step 3	**diameter peer** *peer name* **Example:** `RP/0/RSP0/CPU0:router(config)# diameter peer GX_SERVER`	Configures DIAMETER peer.
Step 4	**transport security-type tls** **Example:** `RP/0/RSP0/CPU0:router(config-dia-peer)# transport security-type tcp`	[Optional] Configures the DIAMETER security type as **TLS**.
Step 5	**transport tcp port** *port_num* **Example:**	Configures the DIAMETER transport protocol used for establishing the connection with the peer, along with the port number (Optional) that the remote peer uses for DIAMETER messages.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-dia-peer)# transport tcp port 3868`	Currently only TCP is supported as DIAMETER transport protocol.
Step 6	**destination host** *host_string* **Example:** `RP/0/RSP0/CPU0:router(config-dia-peer)# destination host dcca1.cisco.com`	Configures the hostname of the peer in Fully Qualified Domain Name (FQDN) format. This value is sent in various messages so that intermediate proxies can correctly route the packets.
Step 7	**destination realm** *realm_string* **Example:** `RP/0/RSP0/CPU0:router(config-dia-peer)# destination realm GX_REALM`	[Optional] Configures the realm to which the peer belongs to. The **destination realm** is added by AAA clients while sending a request to AAA server, using the AAA_AT_DESTINATION_REALM attribute. If this attribute is not present, then the realm information is retrieved using the **User name** field. If the clients do not add the attribute, then the value configured in the peer mode is used while sending messages to the destination peer.
Step 8	**address** [**ipv4** | **ipv6**] *ip_addr* **Example:** `RP/0/RSP0/CPU0:router(config-dia-peer)# address ipv4 2.2.2.2`	Configures IP address of the DIAMETER peer.
Step 9	**ip vrf forwarding** *vrf_table_name* **Example:** `RP/0/RSP0/CPU0:router(config-dia-peer)# ip vrf forwarding VRF1`	[Optional] Configures the VRF associated with the peer, to establish connections with the peers immediately after configuring the peers. If this command is not configured, then the global routing table is used for establishing the connection with the peer. If the VRF associated with the name is not configured, then an error message mentioning that is displayed, and this command does not have any effect.
Step 10	**source-interface** *intf-type intf-name* **Example:** `RP/0/RSP0/CPU0:router(config-dia-peer)# source-interface Bundle-Ether 1`	[Optional] Configures the source-interface to be used for the DIAMETER connection. The diameter client uses this source address and port to initiate the TCP connection to the peer. This command is also available in global configuration mode, when used with **diameter** keyword.
Step 11	**peer-type server** **Example:** `RP/0/RSP0/CPU0:router(config-dia-peer)# peer-type server`	Configures the peer type. By default, the peer type is, **server**.

	Command or Action	Purpose
Step 12	**root** **Example:** `RP/0/RSP0/CPU0:router(config-dia-peer)# root`	Returns the configuration mode back to the global configuration mode.
Step 13	**diameter origin host** *host-name* **Example:** `RP/0/RSP0/CPU0:router(config)# diameter origin host 1.1.1.1`	Configures the origin host information. The origin host information is sent in different requests to the DIAMETER peer and it maps to multiple IP addresses. If this value is not configured, then a NULL string is sent. Therefore, this is a mandatory configuration.
Step 14	**diameter origin realm** *realm-string* **Example:** `RP/0/RSP0/CPU0:router(config)# diameter origin realm cisco.com`	[Optional] Configures the origin realm information. The origin realm information is sent in each request to the DIAMETER peer. If this value is not configured, then a NULL string is sent. Therefore, this is a mandatory configuration.
Step 15	**diameter timer** [**connection** | **transaction** | **watchdog**] *timer-value* **Example:** `RP/0/RSP0/CPU0:router(config)# diameter timer watchdog 300`	Configures global timers for DIAMETER. • **Connection** timer is used to delay the connection establishment or re-establishment of client with the DIAMETER server. It determines the frequency of transport connection attempts with the peer when there is no active connection with the peer. • **Transaction** timer is used for setting the frequency of transaction attempts. That is, the duration for which the client waits for any response message from the peer. • **Watchdog** timer is used to periodically send the Device-Watch-Dog to the DIAMETER server to test the link status. **Note** These timers can also be configured at the peer level (in diameter peer configuration mode). By default, the peers inherit the globally configured timer values. But, if the timer values are configured at peer level as well, then the peer level timer values take precedence over the globally configured timer values.
Step 16	**diameter vendor supported** [**cisco** | **etsi** | **threegpp** | **vodafone**] **Example:** `RP/0/RSP0/CPU0:router(config)# diameter vendor supported cisco`	Advertises the various vendor AVPs that the DIAMETER node understands. This information is passed to the peer in capability exchange messages.

	Command or Action	Purpose
Step 17	**diameter tls trustpoint** *label* **Example:** `RP/0/RSP0/CPU0:router(config)# diameter tls trustpoint DIAMETER_TRUSTPOINT`	Specifies the **trustpoint** name to be used in the certificate to be used for DIAMETER TLS exchange. If a **trustpoint** name is not provided, then the default **trustpoint** is used.

Step 18	**diameter** {**gx** | **gy**} [**retransmit** *retansmit-timer-val* | **tx-timer** *tx-timer-val*]
	Example:
	`RP/0/RSP0/CPU0:router(config)# diameter gx retransmit 5`
	`RP/0/RSP0/CPU0:router(config)# diameter gx tx-timer 100`
		Configures the re-transmit and the transaction timers for Gx and Gy applications.

Step 19	**commit**	

Configuring DIAMETER Connection in BNG: Example

```
DIAMETER-specific configurations:

diameter gx
diameter gy
diameter peer GX_SERVER
 destination realm GX_REALM
 address ipv4 2.2.2.2
!
diameter peer GY_SERVER
 transport tcp port 3869
 destination realm GY_REALM
 address ipv4 2.2.2.2
!
diameter peer NASREQ_SERVER
 address ipv4 1.1.1.2
!
diameter timer watchdog 300
diameter origin host 1.1.1.1
diameter origin realm cisco.com
diameter vendor supported threegpp
diameter vendor supported cisco
diameter vendor supported vodafone
```

Configuring AAA for DIAMETER Peer in BNG

Perform this task to configure AAA for DIAMETER NASREQ application in BNG router.

Before you begin

Prior to this task, you must set up the DIAMETER peer in BNG router. For details, see Configuring DIAMETER Peer in BNG, on page 306.

SUMMARY STEPS

1. **configure**
2. **aaa group server** {**diameter** | **radius**} *server-group-name*
3. **server** *peer_name*
4. **aaa authentication subscriber** {*list-name* | **default**} **group** {*server-group-name* | **diameter** | **radius**}
5. **aaa authorization subscriber** {*list-name* | **default**} **group** {*server-group-name* | **diameter** | **radius**}
6. **aaa accounting subscriber** {*list-name* | **default**} **group** {*server-group-name* | **diameter** | **radius**}
7. **aaa accounting service** {*list-name* | **default**} **group** {*server-group-name* | **diameter** | **radius**}
8. **aaa authorization policy-if** {*list-name* | **default**} **group** {*server-group-name* | **diameter** | **radius**}
9. **aaa authorization prepaid** {*list-name* | **default**} **group** {*server-group-name* | **diameter** | **radius**}
10. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**aaa group server** {**diameter** | **radius**} *server-group-name* **Example:** `RP/0/RSP0/CPU0:router(config)# aaa group server diameter GX_SG`	Configures the named server group for DIAMETER, and enters the server group sub-mode.
Step 3	**server** *peer_name* **Example:** `RP/0/RSP0/CPU0:router(config-sg-diameter)# server GX_SERVER`	Attaches the globally configured DIAMETER server (configured using **diameter peer** command) having the same name, to the server group. If a server is not configured with the same name, then an error message mentioning that is displayed. Unlike for RADIUS, DIAMETER does not have private servers. DIAMETER considers a server that does not have a VRF name configured, as a global server, and it uses global routing table for that particular server.
Step 4	**aaa authentication subscriber** {*list-name* | **default**} **group** {*server-group-name* | **diameter** | **radius**} **Example:** `RP/0/RSP0/CPU0:router(config)# aaa authentication subscriber default group diameter`	Configures subscriber authentication with DIAMETER protocol using NASREQ application.

	Command or Action	Purpose
Step 5	**aaa authorization subscriber** {*list-name* | **default**} **group** {*server-group-name* | **diameter** | **radius**} **Example:** `RP/0/RSP0/CPU0:router(config)# aaa authorization subscriber default group diameter`	Configures subscriber authorization with DIAMETER protocol using NASREQ application.
Step 6	**aaa accounting subscriber** {*list-name* | **default**} **group** {*server-group-name* | **diameter** | **radius**} **Example:** `RP/0/RSP0/CPU0:router(config)# aaa accounting subscriber default group diameter`	Configures subscriber session accounting to DIAMETER server using Base Accounting Application.
Step 7	**aaa accounting service** {*list-name* | **default**} **group** {*server-group-name* | **diameter** | **radius**} **Example:** `RP/0/RSP0/CPU0:router(config)# aaa accounting service default group diameter`	Configures to carry subscriber service accounting records to DIAMETER server using Base Accounting Application.
Step 8	**aaa authorization policy-if** {*list-name* | **default**} **group** {*server-group-name* | **diameter** | **radius**} **Example:** `RP/0/RSP0/CPU0:router(config)# aaa authorization policy-if policy_meth group GX_SG`	Configures authorization lists for policy interface (Gx interface).
Step 9	**aaa authorization prepaid** {*list-name* | **default**} **group** {*server-group-name* | **diameter** | **radius**} **Example:** `RP/0/RSP0/CPU0:router(config)# aaa authorization prepaid prepaid_meth group GY_SG`	Configures authorization lists for prepaid (Gy interface).
Step 10	**commit**	

Configuring AAA for DIAMETER Connection in BNG: Example

```
AAA configurations:

aaa group server diameter GX_SG
 server GX_SERVER
!
aaa group server diameter GY_SG
 server GY_SERVER
!
aaa group server diameter NASREQ_SG
 server NASREQ_SERVER
!
```

```
aaa authorization network default group radius
aaa accounting service default group radius
aaa accounting subscriber default group radius
aaa accounting subscriber nasreq_acct_list group NASREQ_SG
aaa authorization subscriber default group radius
aaa authorization subscriber nasreq_author_list group NASREQ_SG
aaa authorization policy-if policy_meth group GX_SG
aaa authentication subscriber default group radius
aaa authorization prepaid prepaid_meth group GY_SG

Prepaid Service:

dynamic-template
 type service prepaid
  service-policy input qos_in_parent1 merge 10 acct-stats
  service-policy output qos_out_parent1 merge 10 acct-stats
  accounting aaa list default type service periodic-interval 30
  prepaid-config prepaid_config

Prepaid Template:

subscriber
 accounting prepaid prepaid_config
  threshold volume 100
  method-list authorization prepaid_meth
  threshold time 100
  password cisco

Policy Map:

policy-map type control subscriber diam_policy
 event session-start match-first
  class type control subscriber dual-stack do-until-failure
   10 activate dynamic-template DYN_TEMP_IPSUB_DUAL
   20 authorize aaa list default identifier source-address-mac password welcome
   30 authorize aaa list policy_meth identifier username password welcome
  !
 !
 end-policy-map
!
```

Verification of DIAMETER Configurations in BNG

These show commands can be used to verify the DIAMETER configurations in BNG:

SUMMARY STEPS

1. **show tcp brief**
2. **show diameter peer**
3. **show diameter gx statistics**
4. **show diameter gy statistics**
5. **show diameter gx session** *session-id-string*
6. **show diameter gy session** *session-id-string*
7. **show diameter nas session** [**checkpoint** | **session** | **summary**]
8. **show checkpoint dynamic process diameter**

DETAILED STEPS

Step 1 **show tcp brief**

Example:

```
RP/0/RSP0/CPU0:router# show tcp brief

PCB        VRF-ID       Recv-Q Send-Q Local Address   Foreign Address    State
0x1016cc7c 0x60000000     0      0    2.2.2.1:28691   2.2.2.2:3869       ESTAB
0x1016bbc8 0x60000000     0      0    2.2.2.1:24698   2.2.2.2:3868       ESTAB
0x1013ccc0 0x60000000     0      0    0.0.0.0:23      0.0.0.0:0          LISTEN
0x10138db8 0x00000000     0      0    0.0.0.0:23      0.0.0.0:0          LISTEN
```

Displays a summary of the TCP connection table.

Step 2 **show diameter peer**

Example:

```
RP/0/RSP0/CPU0:router# show diameter peer

Origin Host :
Origin Realm :
Source Interface :
TLS Trustpoint :
Connection timer value : 30 seconds
Watchdog timer value : 300 seconds
Transaction timer value : 30 seconds
Number of Peers:3

Peer name : GX_SERVER
        type : SERVER
        Address/port : 2.2.2.2/3868
        Transport protocol : TCP
        Peer security protocol : NONE
        connection timer : 30 seconds
        watchdog timer value : 300 seconds
        transaction timer value : 30 seconds
        VRF name : default
        Source-interface :
        Destination realm : GX_REALM
        Destination host name :
        Peer connection status : Open

      Peer Statistics
------------------------------
          IN     /    OUT
------------------------------
ASR       0             0
ASA       0             0
ACR       0             0
ACA       0             0
CER       0             1
CEA       1             0
DWR       0             0
DWA       0             0
DPR       0             0
DPA       0             0
RAR       0             0
RAA       0             0
```

```
STR       0             0
STA       0             0
AAR       0             0
AAA       0             0
CCR       0             0
CCA       0             0
Malformed Rcvd   :  0
Prot. Errs Sent  :  0                Prot. Errs Rcvd  :  0
Trans. Errs Sent :  0                Trans. Errs Rcvd :  0
Perm. Errs Sent  :  0                Perm. Errs Rcvd  :  0
```

Displays DIAMETER peer information.

Step 3 **show diameter gx statistics**

Example:

```
RP/0/RSP0/CPU0:router# show diameter gx statistics
CCR Initial Messages                : 1
CCR Initial Messages Sent Failed    : 0
CCR Initial Messages Timed Out      : 0
CCR Initial Messages Retry          : 0
CCR Update Messages                 : 0
CCR Update Messages Sent Failed     : 0
CCR Update Messages Timed Out       : 0
CCR Update Messages Retry           : 0
CCR Terminate Messages              : 0
CCR Terminate Messages Sent Failed  : 0
CCR Terminate Messages Timed Out    : 0
CCR Terminate Messages Retry        : 0
CCA Initial Messages                : 1
CCA Initial Messages Error          : 0
CCA Update Messages                 : 0
CCA Update Messages Error           : 0
CCA Terminate Messages              : 0
CCA Terminate Messages Error        : 0
RAR Received Messages               : 0
RAR Received Messages Error         : 0
RAA Sent Messages                   : 0
RAA Sent Messages Error             : 0
ASR Received Messages               : 0
ASR Received Messages Error         : 0
ASA Sent Messages                   : 0
ASA Sent Messages Error             : 0
Session Termination Messages Recvd  : 0
Unknown Request Messages            : 0
Restored Sessions                   : 0
Total Opened Sessions               : 1
Total Closed Sessions               : 0
Total Active Sessions               : 1
```

Displays DIAMETER gx statistics.

Step 4 **show diameter gy statistics**

Example:

```
RP/0/RSP0/CPU0:router# show diameter gy statistics

CCR Initial Messages                : 1
CCR Initial Messages Sent Failed    : 0
```

```
CCR Initial Messages Timed Out      : 0
CCR Initial Messages Retry          : 0
CCR Update Messages                 : 4
CCR Update Messages Sent Failed     : 0
CCR Update Messages Timed Out       : 0
CCR Update Messages Retry           : 0
CCR Terminate Messages              : 1
CCR Terminate Messages Sent Failed  : 0
CCR Terminate Messages Timed Out    : 0
CCR Terminate Messsages Retry       : 0
CCA Initial Messages                : 1
CCA Initial Messages Error          : 0
CCA Update Messages                 : 4
CCA Update Messages Error           : 0
CCA Terminate Messages              : 1
CCA Terminate Messages Error        : 0
RAR Received Messages               : 0
RAR Received Messages Error         : 0
RAA Sent Messages                   : 0
RAA Sent Messages Error             : 0
ASR Received Messages               : 0
ASR Received Messages Error         : 0
ASA Sent Messages                   : 0
ASA Sent Messages Error             : 0
Unknown Request Messages            : 0
Restored Sessions                   : 0
Total Opened Sessions               : 2
Total Closed Sessions               : 1
Total Active Sessions               : 1
```

Displays DIAMETER gy statistics.

Step 5 **show diameter gx session** *session-id-string*

Example:

```
RP/0/RSP0/CPU0:router# show diameter gx session 461419

Gx Session Status for  [461419]
        Session Status        : ACTIVE
        Diameter Session ID   : 1.1.1.1;4;461419;1185991
        Gx Session State      : OPEN
        Request Number        : 0
        Request Type          : INITIAL REQUEST
        Request Retry Count   : 0
```

Displays DIAMETER gx session information.

Step 6 **show diameter gy session** *session-id-string*

Example:

```
RP/0/RSP0/CPU0:router# show diameter gy session 461421

Gy Session Status for  [461421]
        Session Status        : ACTIVE
        Diameter Session ID   : 1.1.1.1;4;461421;1186625
        Gy Session State      : OPEN
        Request Number        : 1
        Request Type          : UPDATE REQUEST
```

```
        Request Retry Count   : 0
```

Displays DIAMETER gy session information.

Step 7 **show diameter nas session** [**checkpoint** | **session** | **summary**]

Example:

```
RP/0/RSP0/CPU0:router# show diameter nas session

Gy Session Status for  [461421]
        Session Status        : ACTIVE
        Diameter Session ID   : 1.1.1.1;4;461421;1186625
        Gy Session State      : OPEN
        Request Number        : 1
        Request Type          : UPDATE REQUEST
        Request Retry Count   : 0

RP/0/RSP0/CPU0:router# show diameter nas session 00070a6f

Nas Session status for [00070a6f]
        Session Status            :  Active
        Diameter Session ID       : 1.1.1.1;4;461423;1187179

        Authentication Status     : NA
        Authorization Status      : SUCCESS
        Accounting Status (Start) : NA
        Accounting Status (Stop)  : NA
        Disconnect status         : NA

  Peer Information :
        Server group     :  NASREQ_SG
        Server Used      :  NASREQ_SERVER

RP/0/RSP0/CPU0:router# show diameter nas summary

NAS Statistics :

    NAS Initiated msgs :

        Authentication       ::

          In                   :          0   Out                 :           0
          Requests received    :          0   Requests send       :           0
          Response received    :          0   Result forwaded     :           0
          Transaction Succeeded:          0   Transactions Failed  :          0

        Authorization        ::

          In                   :          1   Out                 :           1
          Requests received    :          1   Requests send       :           1
          Response received    :          1   Result forwaded     :           1
          Transaction Succeeded:          1   Transactions Failed  :          0

        Accounting (Start)   ::

          In                   :          0   Out                 :           0
          Requests received    :          0   Requests send       :           0
          Response received    :          0   Result forwaded     :           0
          Transaction Succeeded:          0   Transactions Failed  :          0

        Accounting (Stop)    ::
```

```
          In                    :           0   Out                 :           0
          Requests received     :           0   Requests send       :           0
          Response received     :           0   Result forwaded     :           0
          Transaction Succeeded:            0   Transactions Failed  :          0

        Accounting (Interim) ::

          In                    :           0   Out                 :           0
          Requests received     :           0   Requests send       :           0
          Response received     :           0   Result forwaded     :           0
          Transaction Succeeded:            0   Transactions Failed  :          0

        Disconnect           ::

          In                    :           0   Out                 :           0
          Requests received     :           0   Requests send       :           0
          Response received     :           0   Result forwaded     :           0
          Transaction Succeeded:            0   Transactions Failed  :          0


    Server Initiated msgs :

        Coa  (RAR)           ::

          In                    :           0   Out                 :           0
          Requests received     :           0   Requests send       :           0
          Response received     :           0   Result forwaded     :           0
          Transaction Succeeded:            0   Transactions Failed  :          0

        POD  (ASR)           ::

          In                    :           0   Out                 :           0
          Requests received     :           0   Requests send       :           0
          Response received     :           0   Result forwaded     :           0
          Transaction Succeeded:            0   Transactions Failed  :          0
Diameter NAS summary
```

Displays DIAMETER NAS information.

Step 8 **show checkpoint dynamic process diameter**

Example:

```
RP/0/RSP0/CPU0:router# show checkpoint dynamic process diameter

     Name           Version         ID    Seg #Objects   Length  InfoLen  Flags
---------------------------------------------------------------------------------
0x00000003     0,     0,    0 0x40001c00      M        0      292        4   I M
0x00000004     0,     0,    0 0x40001d00      M        1      264        4   I M
0x00000002     0,     0,    0 0x40001e00      M        1       24        4   I M
0x00000001     0,     0,    0 0x40001f00      M        1       24        4   I M

Segment 0: Number of pages allocated: 4
Segment 0: Number of pages free: 3

Segment 1: Number of pages allocated: 9
Segment 1: Number of pages free: 3

-----------------------------------------------------------------------------------
```

Displays checkpoint information of DIAMETER process.

Additional References

These sections provide references related to implementing DIAMETER.

RFCs and Standards

Standard/RFC	
RFC-6733	Diameter Base Protocol
RFC-4006	Diameter Credit-Control Application
RFC-4005	Diameter Network Access Server Application (NASREQ)
RFC-3046	DHCP Relay Agent Information Option
RFC-3539	Authentication, Authorization and Accounting (AAA) Transport Profile
3GPP TS 129 212 V11.10.0	Universal Mobile Telecommunications System (UMTS); LTE; Policy and Charging Control (PCC); Reference Points for Gx interface support.
3GPP TS 132 299 V11.9.1	Technical Specification on Diameter charging applications used for Gx and Gy interface support.

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

APPENDIX A

XML Support for BNG Features

Most BNG features, such as AAA, DHCP, Policy Plane, PPPoE, DAPS, and Subscriber Database support XML based router configuration. The Cisco XML API can be used to configure routers or request information about configuration, management, and operation of the routers. For details about using the Cisco XML API, see the latest release of *Cisco IOS XR XML API Guide* listed at http://www.cisco.com/en/US/products/ps9853/products_programming_reference_guides_list.html.

The Cisco XML API uses XML commands to configure the router. The following sections list the supported XML commands for the BNG features.

AAA XML Support

The support for XML is available for RADIUS that retrieves the accounting and authorization request statistics. The mapping between CLI and XML entries for the AAA commands are as follows:

CLI	XML
radius-server dead-criteria time	`AAA.RADIUS. DeadCriteria.Time`
radius-server dead-criteria tries	`AAA.RADIUS. DeadCriteria.Tries`
radius-server ipv4 dscp <value>	`AAA.RADIUS. IPv4.DSCP`
radius-server key {0 | 7 | LINE}	`AAA.RADIUS.Key`
radius-server retransmit <limit>	`AAA.RADIUS.Retransmit`

CLI	XML
radius-server timeout <number>	`AAA.RADIUS.Timeout`
radius-server source-port extended	`AAA.RADIUS.SourcePort.Extended`
radius-server deadtime	`AAA.RADIUS.DeadTime`
radius-server load-balance method least-outstanding	`AAA.RADIUS.LoadBalance.Method.LeastOutstanding`
radius-server attribute list <attribute-name>	`AAA.RADIUS.AttributeListTable.AttributeList.Enable`
radius-server attribute list <attribute-name> attribute <radius-attributes>	`AAA.RADIUS.AttributeListTable.AttributeList.Attribute`
radius-server vsa attribute ignore unknown	`AAA.RADIUS.VSA.Attribute.Ignore.Unknown`
Radius-server host <> retransmit	`AAA.RADIUS.HostTable.Host.Retransmit`
Radius-server host <> timeout	`AAA.RADIUS.HostTable.Host.Timeout`
radius-server host <> key {0 | 7 | LINE}	`AAA.RADIUS.HostTable.Host.Key`
aaa server radius dynamic-author client <ip-address> vrf <vrf-name> server-key {0 | 7 | LINE}	`AAA.RADIUS.DynamicAuthorization.ClientTable.Client.ServerKey`
aaa server radius dynamic-author ignore {server key | session key }	`AAA.RADIUS.DynamicAuthorization.Ignore`
aaa server radius dynamic-author port <port num>	`AAA.RADIUS.DynamicAuthorization.Port`

CLI	XML
aaa accounting system default start-stop [broadcast] {group {radius | NAME1}} [group NAME2..] aaa accounting system rp-failover default start-stop [broadcast] {group {radius | NAME1}} [group NAME2..	`AAA.AccountingTable.Accounting`
aaa radius attribute nas-port-id format FORMAT_NAME	`AAA.RADIUSAttribute.NASPortID.Format`
aaa group server radius <group-name> { authorization } { reply | reject} <name>	`AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.Authorization.Reply`
aaa group server radius <group-name> { authorization} { accept | request } <name>	`AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.Authorization.Request`
aaa group server radius <group-name> { accounting } { accept | request} <name>	`AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.Accounting.Request`
aaa group server radius <group-name> { accounting } { reply | reject} <name>	`AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.Accounting.Reply`
aaa group server radius <group-name> load-balance	`AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.LoadBalance.Method.LeastBounding`

CLI	XML
method least-bounding	
aaa group server radius group1 source-interface	AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.SourceInterface
aaa group server radius <radius-group> vrf <>	AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.VRF
aaa group server radius <radius-group> deadtime <>	AAA.ServerGroups.RADIUSServerGroupTable.RADIUSServerGroup.DeadTime
aaa group server radius <> server-private <host>	AAA.ServerGroups.RADIUSGroupTable.RADIUSGroup.PrivateServerTable.PrivateServer
show radius accounting	RADIUS.Accounting
show radius authentication	RADIUS.Authentication
show radius client	RADIUS.Client
show radius dynamic-author	RADIUS.DynamicAuthorization
show radius dead-criteria host <ip>	RADIUS.DeadCriteria.HostTable.Host
show radius server-groups	RADIUS.ServerGroups

DHCP XML Support

The support for XML is available for DHCP that retrieves the client bindings, profile information, and DHCPv4 proxy statistics. It allows the management clients to perform client bindings based on Circuit-ID, Remote-ID, Mac-Address, user profile information, and DHCPv4 proxy statistics. The mapping between CLI and XML entries for the DHCP commands are as follows:

CLI	XML
dhcp ipv4 profile <name> proxy relay information check	DHCPv4.ProfileTable.Profile.Proxy.RelayInformation.Check

XML Support for BNG Features

<table>
<tr><th>CLI</th><th>XML</th></tr>
<tr><td>dhcp ipv4 profile <name>proxy relay information option[vpn | allow-untrusted | remote-id <name>]</td><td>DHCPv4.ProfileTable.Profile.Proxy.RelayInformation.AllowUntrusted
DHCPv4.ProfileTable.Profile.Proxy.RelayInformation.VPN
DHCPv4.ProfileTable.Profile.Proxy.RelayInformation.RemoteID</td></tr>
<tr><td>dhcp ipv4 interface GigabitEthernet <interface-name> proxy profile <name></td><td>DHCPv4.InterfaceTable.Interface.Proxy.Profile</td></tr>
<tr><td>dhcp ipv4 profile <name>proxy relay information policy [drop | keep | replace]</td><td>DHCPv4.ProfileTable.Profile.Proxy.RelayInformation.Policy</td></tr>
<tr><td>dhcp ipv4 profile <name>proxy helper-address [vrf <name>] <server-ip-addr> [giaddr <ip-addr>]</td><td>DHCPv4.ProfileTable.Profile.Proxy.VRFTable.VRF.HelperAddressTable.HelperAddress</td></tr>
<tr><td>dhcp ipv4 profile <name> proxy broadcast-flag policy check</td><td>DHCPv4.ProfileTable.Profile.Proxy.BroadcastFlag.Policy</td></tr>
<tr><td>dhcp ipv4 profile <name>proxy class <class-name> helper-address [vrf <name>] <server-ip-addr> [giaddr <ip-addr>] match vrf <name> match option [124 | 125 | 60 | 77] hex <value> [mask <value>]</td><td>DHCPv4.ProfileTable.Profile.Proxy.ClassTable.Class
DHCPv4.ProfileTable.Profile.Proxy.ClassTable.Class.VRFTable.VRF.HelperAddressTabl
DHCPv4.ProfileTable.Profile.Proxy.ClassTable.Class.Match.VRF
DHCPv4.ProfileTable.Profile.Proxy.ClassTable.Class.Match.Option</td></tr>
<tr><td>dhcp ipv4 interface <interface> none</td><td>DHCPv4.InterfaceTable.Interface.None</td></tr>
</table>

CLI	XML
dhcp ipv4 interface <interface> proxy [information option format-type circuit-id <cir-id>]	`DHCPv4.InterfaceTable.Interface.Proxy.CircuitID`
dhcp ipv4 vrf vrfname proxy profile <name>	`DHCPv4.VRFTable.VRF`
show dhcp ipv4 proxy binding circuit-id <cid> location <locationSpecifier>	`DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable[DHCPv4ProxyCircuitIDFilter(Naming C`
show dhcp ipv4 proxy binding remote-id <rid> location <locationSpecifier>	`DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable[DHCPv4ProxyRemoteIDFilter(Naming Re`
show dhcp ipv4 proxy binding interface <ifSpecifier>	`DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable[DHCPv4ProxyInterfaceFilter(Naming Int`
show dhcp ipv4 proxy binding mac-address <addr> location <locationSpecifier>	`DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable[DHCPv4ProxyMACAddressFilter(Naming`
show dhcp ipv4 proxy binding location <locationSpecifier>	`DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable[DHCPv4ProxyBriefFilter]`
show dhcp ipv4 proxy binding detail location <locationSpecifier>	`DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable.Client`
show dhcp ipv4 proxy binding summary location <locationSpecifier>	`DHCPv4.NodeTable.Node.Proxy.Binding.Summary`
show dhcp ipv4 proxy binding vrf <vrfname>	`DHCPv4.NodeTable.Node.Proxy.Binding.ClientTable[DHCPv4PProxyVRFFilter(Naming VRFNam`

CLI	XML
show dhcp ipv4 proxy profile name <profile-name> location <locationSpecifier>	`DHCPv4.NodeTable.Node.Proxy.ProfileTable.Profile`
show dhcp vrf <name> ipv4 proxy statistics location <locationSpecifier>	`DHCPv4.NodeTable.Node.Proxy.VRFTable.VRF.Statistics`
show dhcp ipv4 proxy statistics [location < loc >]	`DHCPv4.NodeTable.Node.Proxy.Statistics`

Control Policy XML Support

The support for XML is available for policy plane that retrieves subscriber management and subscriber session related information. The mapping between CLI and XML entries for the control policy commands are as follows:

CLI	XML
interface <intf> service-policy type control subscriber <policy-name>	`InterfaceConfigurationTable.InterfaceConfiguration.ControlSubscriber.ServicePolicy`
sh sub sess all loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable`
sh sub sess all detail loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberDetailAllSessionFilter)`
sh sub sess all summary loc <loc>	`Subscriber.Session.NodeTable.Node.Summary`
sh sub sess all username loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberAllUsenameFilter)`
sh sub sess filter interface	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberInterfaceBriefFilter) {Naming InterfaceName}`

CLI	XML
<intf-name> loc <loc>	
sh sub sess filter interface <intf-name> detail loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberInterfaceDetailFilter) {Naming InterfaceName}`
sh sub sess filter ipv4-address <IPv4-addr> loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberIPv4AddressVRFDetailFilter) {Naming VRF Name, Address}`
sh sub sess filter ipv4-address <IPv4-addr> detail loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberIPv4AddressVRFBriefFilter) {Naming VRF Name, Address}`
sh sub sess filter mac-address <mac-addr> loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberMACAddressBriefFilter) {Naming MACAddress}`
sh sub sess filter mac-address <mac-addr> detail loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberMACAddressDetailFilter) {Naming MACAddress}`
sh sub sess filter state <state> loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberStateBriefFilter) {Naming State}`
sh sub sess filter state <state> detail loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberStateDetailFilter) {Naming State}`
sh sub sess filter username <uname> loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberUsernameBriefFilter) {Naming Username}`
sh sub sess filter	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberUsernameDetailFilter) {Naming Username}`

CLI	XML
username <uname> detail loc <loc>	
sh sub sess filter ipv4-address <IPv4 addr> vrf <vrf> loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberIPv4AddressVRFBriefFilter) {Naming VRF Name, Address}`
sh sub sess filter ipv4-address <IPv4-addr> vrf <vrf> detail loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberIPv4AddressVRFDetailFilter) {Naming VRF Name, Address}`
sh sub sess filter vrf <vrf-name> loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberIPv4AddressVRFBriefFilter) {Naming VRF Name, Address }`
sh sub sess filter vrf <vrf-name> detail loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable(SubscriberIPv4AddressVRFDetailFilter) {Naming VRF Name, Address }`
sh sub sess sub-label <0-ffffffff> loc <loc>	`Subscriber.Session.NodeTable.Node.SessionTable.Session{Naming SessionID}`
sh sub man stat AAA accounting loc <loc>	`Subscriber.Manager.NodeTable.Node.Statistics.AAA.Accounting`
sh sub man stat AAA accounting total loc <loc>	`Subscriber.Manager.NodeTable.Node.Statistics.AAA.AggregateAccounting`
sh sub man stat AAA authentication loc <loc>	`Subscriber.Manager.NodeTable.Node.Statistics.AAA.Authentication`
sh sub man stat AAA	`Subscriber.Manager.NodeTable.Node.Statistics.AAA.AggregateAuthentication`

CLI	XML
authentication total loc <loc>	
sh sub man stat AAA authorization loc <loc>	`Subscriber.Manager.NodeTable.Node.Statistics.AAA.Authorization`
sh sub man stat AAA authorization total loc <loc>	`Subscriber.Manager.NodeTable.Node.Statistics.AAA.AggregateAuthorization`
sh sub man stat AAA COA loc <loc>	`Subscriber.Manager.NodeTable.Node.Statistics.AAA.ChangeOfAuthorization`
sh sub man stat AAA COA total loc <loc>	`Subscriber.Manager.NodeTable.Node.Statistics.AAA.AggregateChangeOfAuthorization`
sh sub man stat AAA all loc <loc>	`Subscriber.Manager.NodeTable.Node.Statistics.AAA`
sh sub man stat AAA all total loc <loc>	`Subscriber.Manager.NodeTable.Node.Statistics.AAA`
sh sub man stats summary total <loc>	`Subscriber.Manager.NodeTable.Node.Statistics.AggregateSummary`

DAPS XML Support

The support for XML is available for distributed address pool service (DAPS) that retrieves the pool parameters for distributed address pool services, and allows the management clients to get number of free, allocated and excluded addresses based on VRF and pool name. The mapping between CLI and XML entries for the DAPS commands are as follows:

CLI	XML
pool vrf <vrf-name> ipv4 <poolname>pool ipv4 <poolname>	`PoolService.VRFTable.VRF.IPv4.Pool.Enable`
pool vrf <VRFName> ipv4 <PoolName> * address-range <RangeStart> <RangeEnd>pool ipv4 <PoolName> * address-range <RangeStart> <RangeEnd>	`PoolService.VRFTable.VRF.IPv4.Pool.AddressRangeTable.AddressRange`

CLI	XML
pool vrf <VRFName> ipv4 <PoolName> * exclude <RangeStart> <RangeEnd>pool vrf <VRFName> ipv4 <PoolName> * exclude <RangeStart> <RangeEnd>pool ipv4 <PoolName> * exclude <RangeStart> <RangeEnd>	`PoolService.VRFTable.VRF.IPv4.Pool.ExcludeTable.Exclude`
Pool vrf <VRFName> ipv4 <PoolName> utilization-mark high <>pool ipv4 <PoolName> utilization-mark high <>	`PoolService.VRFTable.VRF.IPv4.Pool.UtilizationMark.High`
Pool vrf <VRFName> ipv4 <PoolName> utilization-mark low <>pool ipv4 <PoolName> utilization-mark low <>	`PoolService.VRFTable.VRF.IPv4.Pool.UtilizationMark.Low`
show pool vrf <vrf-name> ipv4	`PoolService.NodeTable.Node.VRFTable.VRF.IPv4`
show pool ipv4 name <poolname>	`PoolService.NodeTable.Node.PoolTable.Pool.IPv4.Detail`
show pool ipv4 name <poolname> verbose	`PoolService.NodeTable.Node.PoolTable.Pool.IPv4.Verbose`
show pool ipv4show pool vrf all ipv4	`PoolService.NodeTable.Node.VRFTable`

PPPoE XML Support

XML support is available for PPP over Ethernet (PPPoE) sessions. The mapping between CLI and XML entries for the PPPoE feature commands are:

CLI	XML
pado delay {<delay>}	`set PadoDelay.Default {<delay>}`
pado delay circuit-id {<delay>}	`set PadoDelay.CircuitId {<delay>}`
pado delay remote-id {<delay>}	`set PadoDelay.RemoteId {<delay>}`
pado delay circuit-id string {<string>} {<delay>}	`set PadoDelay.CircuitIdString{<string>} {<delay>}`
pado delay circuit-id contains {<string>} {<delay>}	`set PadoDelay.CircuitIdSubString{<string>} {<delay>}`
pado delay remote-id string {<string>} {<delay>}	`set PadoDelay.RemoteIdString{<string>} {<delay>}`
pado delay remote-id contains {<string>} {<delay>}	`set PadoDelay.RemoteIdSubString{<string>} {<delay>}`
pado delay service-name string {<string>} {<delay>}	`set PadoDelay.ServiceNameString{<string>} {<delay>}`
pado delay service-name contains {<string>} {<delay>}	`set PadoDelay.ServiceNameSubString{<string>} {<delay>}`
pppoe session-id space flat	`set SessionIDSpaceFlat {TRUE}`
pppoe bba-group {<group-name>}	`PPPoECfg.BBAGroup {<group-name>}`

XML Support for BNG Features

CLI	XML
pppoe enable bba-group {<group-name>}	`set PPPoE.EnableBBAGroup {<group-name>}`
ac name {<name>}	`set Tags.ACName {<name>}`
service name {<name>}	`set Tags.ServiceName(<name>).ServiceNameConfigured`
service selection disable	`set Tags.ServiceSelectionDisable`
tag ppp-max-payload deny	`set Tags.PPPMaxPayloadDeny`
tag ppp-max-payload minimum {<min>} maximum {<max>}	`set Tags.PPPMaxPayload {<min>,<max>}`
mtu {<mtu>}	`set MTU {<mtu>}`
sessions max limit {limit} threshold {<threshold>}	`set Sessions.MaxLimit {<limit>,<threshold>}`
sessions access-interface limit {<count>} [threshold {<threshold>}]	`set Sessions.AccessInterfaceLimit {<count>,<threshold>}`
sessions mac limit {<count>} [threshold {<threshold>}]	`set Sessions.MacLimit {<count>,<threshold>}`
sessions mac-iwf limit {<count>} [threshold {<threshold>}]	`set Sessions.MacIWFLimit {<count>,<threshold>}`
sessions mac access-interface limit {<count>} [threshold {<threshold>}]	`set Sessions.MacAccessInterfaceLimit {<count>,<threshold>}`
sessions mac-iwf access-interface limit {<count>} [threshold {<threshold>}]	`set Sessions.MacIWFAccessInterfaceLimit {<count>,<threshold>}`
sessions circuit-id limit {<count>} [threshold {<threshold>}]	`set Sessions.CircuitIDLimit {<count>,<threshold>}`
sessions remote-id limit {<count>} [threshold {<threshold>}]	`set Sessions.RemoteIDLimit {<count>,<threshold>}`
sessions circuit-id-and-remote-id limit {<count>} [threshold {<threshold>}]	`set Sessions.CircuitIDAndRemoteIDLimit {<count>,<threshold>,<radius-override>}`
sessions inner-vlan limit {<count>} [threshold {<threshold>}]	`set Sessions.InnerVLANLimit {<count>,<threshold>}`
sessions mac throttle {<request-count> <request-period> <blocking-period>}	`set Sessions.MacThrottle {<request-count>,<request-period>,<blocking-period>}`
sessions mac access-interface throttle {<request-count> <request-period> <blocking-period>}	`set Sessions.MacAccessInterfaceThrottle {<request-count>,<request-period>,<blocking-period>}`
sessions mac-iwf access-interface throttle {<request-count> <request-period> <blocking-period>}	`set Sessions.MacIWFAccessInterfaceThrottle {<request-count>,<request-period>,<blocking-period>}`
sessions circuit-id throttle {<request-count> <request-period> <blocking-period>}	`set Sessions.CircuitIDThrottle {<request-count>,<request-period>,<blocking-period>}`

CLI	XML
sessions remote-id throttle {<request-count> <request-period> <blocking-period>}	set Sessions.RemoteIDThrottle {<request-count>,<request-period>,<blocking-period>}
sessions circuit-id-and-remote-id throttle {<request-count> <request-period> <blocking-period>}	set Sessions.CircuitIDAndRemoteIDThrottle {<request-count>,<request-period>,<blocking-period>}
sessions inner-vlan throttle {<request-count> <request-period> <blocking-period>}	set Sessions.InnerVLANThrottle {<request-count>,<request-period>,<blocking-period>}
control-packets priority {<cos>}	set ControlPackets.Priority {<cos>}
invalid-session-id drop	set InvalidSessionID {DROP}
invalid-session-id log	set InvalidSessionID {LOG}

Subscriber Database XML Support

The support for XML is available for subscriber database that retrieves the subscriber association and session information and allows the management clients to get subscriber session state, subscriber session information based on unique subscriber label, subscriber association information based on unique subscriber label or interface name or dynamic template name or type. The mapping between CLI and XML entries for the subscriber database commands are as follows:

CLI	XML
show subscriber database association br location <>	Subscriber.Database.NodeTable.Node.Association.LabelTable.Label{Naming SubscriberLabel}
show subscriber database association subscriber-label <> br location<>	Subscriber.Database.NodeTable.Node.Association.LabelTable.Label{Naming SubscriberLabel}
show subscriber database association location <>	Subscriber.Database.NodeTable.Node.Association(SubscriberDatabaseLabelDetailFilter)
show subscriber database association interface-name <> br location<>	Subscriber.Database.NodeTable.Node.Association(SubscriberDatabaseInterfaceBriefFilter){Naming InterfaceName}
show subscriber database association interface-name <> location<>	Subscriber.Database.NodeTable.Node.Association(SubscriberDatabaseInterfaceFilter){Naming InterfaceName}
show subscriber database association type < ipsubscriber |ppp |service-profile |subscriber-service> br location <>	Subscriber.Database.NodeTable.Node.Association(SubscriberDatabaseTemplateTypeBriefFilter){Naming TemplateType}

XML Support for BNG Features

CLI	XML
show subscriber database association type < ipsubscriber |ppp |service-profile |subscriber-service> location <>	`Subscriber.Database.NodeTable.Node.Association(SubscriberDatabaseTemplateTypeFilter){Naming TemplateType}`
show subscriber database session state <all| cfgapply |cfgdone |cfggen |cfgunapply |destroying |error |fatgen |init |sync>	`Subscriber.Database.NodeTable.Node.Session(SubscriberDatabaseSessionStateFilter){Naming Session-State}`
show subscriber database session subscriber-label <> location <>	`Subscriber.Database.NodeTable.Node.Session.LabelTable.Label{Naming SubscriberLabel}`
association subscriber-label <0x0-0xffffffff> brief location R/S/M	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label`
association subscriber-label <0x0-0xffffffff> brief	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label`
association subscriber-label <0x0-0xffffffff> location R/S/M	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label`
association subscriber-label <0x0-0xffffffff>	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label`
association interface-name <ifname> brief location R/S/M	`Subscriber.Database.NodeTable.Node.Association[SubscriberInterfaceBriefFilter(Naming InterfaceName]`
association interface-name <ifname> brief	`Subscriber.Database.NodeTable.Node.Association[SubscriberInterfaceBriefFilter(Naming InterfaceName]`
association interface-name <ifname> location R/S/M	`Subscriber.Database.NodeTable.Node.Association[SubscriberInterfaceFilter(Naming InterfaceName]`
association interface-name <ifname>	`Subscriber.Database.NodeTable.Node.Association[SubscriberInterfaceFilter(Naming InterfaceName]`
association type ppp brief location R/S/M	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type ppp brief	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type ppp location R/S/M	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type ppp	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`

CLI	XML
association type ipsubscriber brief location R/S/M	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type ipsubscriber brief	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type ipsubscriber location R/S/M	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type ipsubscriber	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type subscriber-service brief location R/S/M	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type subscriber-service brief	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type subscriber-service location R/S/M	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type subscriber-service	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type service-profile brief location R/S/M	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type service-profile brief	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type service-profile location R/S/M	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type service-profile	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type user-profile brief location R/S/M	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type user-profile brief	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type user-profile location R/S/M	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association type user-profile	`Subscriber.Database.NodeTable.Node.Association[SubscriberTemplateType(Naming TemplateType]`
association brief location R/S/M	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label`
association brief	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label`

XML Support for BNG Features

CLI	XML
association location R/S/M	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label`
association	`Subscriber.Database.NodeTable.Node.Association.LabelTable.Label`
session subscriber-label <0x0-0xffffffff> location R/S/M	`Subscriber.Database.NodeTable.Node.Session.LabelTable.Label`
session subscriber-label <0x0-0xffffffff>	`Subscriber.Database.NodeTable.Node.Session.LabelTable.Label`
session state init location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state init	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state destroying location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state destroying	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfggen location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfggen	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state fatgen location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state fatgen	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgapply location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgapply	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgdone location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgdone	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgunapply location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgunapply	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgerror location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state cfgerror	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`

CLI	XML
session state error location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state error	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state sync location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state sync	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state all location R/S/M	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`
session state all	`Subscriber.Database.NodeTable.Node.Session[SubscriberSessionStateFilter(Naming State]`

APPENDIX B

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 15: Supported RADIUS IETF Attributes

Name	Value	Type
Acct-Authentic	integer	45
Acct-Delay-Time	integer	41
Acct-Input-Giga-Words	integer	52
Acct-Input-Octets	integer	42

Name	Value	Type
Acct-Input-Packets	integer	47
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-Session-Time	integer	46
Acct-Status-Type	integer	40
Acct-Terminate-Cause	integer	49
CHAP-Challenge	binary	40
CHAP-Password	binary	3
Dynamic-Author-Error-Cause	integer	101
Event-Timestamp	integer	55
Filter-Id	binary	11
Framed-Protocol	integer	7
Framed-IP-Address	ipv4addr	8
Framed-Route	"string"	22
login-ip-addr-host	ipv4addr	14
Multilink-Session-ID	string	50
Nas-Identifier	string	32
NAS-IP-Address	ipv4addr	4
NAS-Port	integer	5
Reply-Message	binary	18
Service-Type	integer	6
Tunnel-Assignment-Id	string	32
Tunnel-Packets-Lost	integer	86
X-Ascend-Client-Primary-DNS	ipv4addr	135
X-Ascend-Client-Secondary-DNS	ipv4addr	136
NAS-IPv6-Address	string	95
Delegated-IPv6-Prefix	binary	123
Stateful-IPv6-Address-Pool	binary	123
Framed-IPv6-Prefix	binary	97

Name	Value	Type
Framed-Interface-Id	binary	96
Framed-IPv6-Pool	string	100
Framed-IPv6-Route	string	99
login-ip-addr-host	string	98

IETF Tagged Attributes on LAC

The IETF Tagged Attributes support on L2TP Access Concentrator (LAC) provides a means of grouping tunnel attributes referring to the same tunnel in an Access-Accept packet sent from the RADIUS server to the LAC. The Access-Accept packet can contain multiple instances of same RADIUS attributes, but with different tags. The tagged attributes support ensures that all attributes pertaining to a given tunnel contain the same value in their respective tag fields, and that each set includes an appropriately-valued instance of the Tunnel-Preference attribute. This conforms to the tunnel attributes that are to be used in a multi-vendor network environment, thereby eliminating interoperability issues among Network Access Servers (NASs) manufactured by different vendors.

For details of RADIUS Attributes for Tunnel Protocol Support, refer RFC 2868.

These examples describe the format of IETF Tagged Attributes:

```
Tunnel-Type = :0:L2TP, Tunnel-Medium-Type = :0:IP, Tunnel-Server-Endpoint = :0:"1.1.1.1",
Tunnel-Assignment-Id = :0:"1", Tunnel-Preference = :0:1, Tunnel-Password = :0:"hello"
```

A tag value of 0 is used in the above example in the format of `:0:`, to group those attributes in the same packet that refer to the same tunnel. Similar examples are:

```
Tunnel-Type = :1:L2TP, Tunnel-Medium-Type = :1:IP, Tunnel-Server-Endpoint = :1:"2.2.2.2",
Tunnel-Assignment-Id = :1:"1", Tunnel-Preference = :1:1, Tunnel-Password = :1:"hello"

Tunnel-Type = :2:L2TP, Tunnel-Medium-Type = :2:IP, Tunnel-Server-Endpoint = :2:"3.3.3.3",
Tunnel-Assignment-Id = :2:"1", Tunnel-Preference = :2:2, Tunnel-Password = :2:"hello"

Tunnel-Type = :3:L2TP, Tunnel-Medium-Type = :3:IP, Tunnel-Server-Endpoint = :3:"4.4.4.4",
Tunnel-Assignment-Id = :3:"1", Tunnel-Preference = :3:2, Tunnel-Password = :3:"hello"

Tunnel-Type = :4:L2TP, Tunnel-Medium-Type = :4:IP, Tunnel-Server-Endpoint = :4:"5.5.5.5",
Tunnel-Assignment-Id = :4:"1", Tunnel-Preference = :4:3, Tunnel-Password = :4:"hello"

Tunnel-Type = :5:L2TP, Tunnel-Medium-Type = :5:IP, Tunnel-Server-Endpoint = :5:"6.6.6.6",
Tunnel-Assignment-Id = :5:"1", Tunnel-Preference = :5:3, Tunnel-Password = :5:"hello"
```

Table 16: Supported IETF Tagged Attributes

IETF Tagged Attribute Name	Value	Type
Tunnel-Type	integer	64
Tunnel-Medium-Type	integer	65
Tunnel-Client-Endpoint	string	66
Tunnel-Server-Endpoint	string	67

IETF Tagged Attribute Name	Value	Type
Tunnel-Password	string	69
Tunnel-Assignment-ID	string	82
Tunnel-Preference	integer	83
Tunnel-Client-Auth-ID	string	90
Tunnel-Server-Auth-ID	string	91

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.

For example, the following AV pair causes Cisco's "multiple named ip address pools" feature to be activated during IP authorization (during PPP's IPCP address assignment):

```
cisco-avpair= "ip:addr-pool=first"
```

If you insert an "*", the AV pair "ip:addr-pool=first" becomes optional. Note that any AV pair can be made optional.

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

```
cisco-avpair= "ip:addr-pool*first"
```

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 17: Supported Cisco Vendor-Specific RADIUS Attributes

Name	Value	Type	Present in AAA message type
access-loop-encapsulation	binary	1	Access-accept, Accounting-request
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
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
acct-policy-in	string	1	Access-request
acct-policy-map	string	1	Access-request
acct-policy-out	string	1	Access-request
actual-data-rate-downstream	integer	1	Access-accept, Accounting-request

Name	Value	Type	Present in AAA message type
actual-data-rate-upstream	integer	1	Access-accept, Accounting-request
actual-interleaving-delay-downstream	integer	1	Access-accept, Accounting-request
actual-interleaving-delay-upstream	integer	1	Access-accept, Accounting-request
addr-pool **Note** This is for IPv4 subscriber.	string	1	Access-accept
addrv6	string	1	Access-accept, Accounting-request
attainable-data-rate-downstream	integer	1	Access-accept, Accounting-request
attainable-data-rate-upstream	integer	1	Access-accept, Accounting-request
circuit-id-tag	string	1	Access-accept, Accounting-request
cisco-nas-port	string	2	Access-accept, Accounting-request
client-mac-address	string	1	Access-accept, Accounting-request
command	string	1	CoA
connect-progress	string	1	Accounting-request
connect-rx-speed	integer	1	Access-accept, Accounting-request
connect-tx-speed	integer	1	Access-accept, Accounting-request
delegated-ipv6-pool	string	1	Access-accept
dhcp-class	string	1	Access-accept
dhcp-client-id	string	1	Accounting-request
dhcp-vendor-class	string	1	Access-request, Accounting-request
dhcpv6-class	string	1	Access-accept
disc-cause-ext	string	1	Accounting-request
disconnect-cause	string	1	Accounting-request
dual-stack-delay	integer	1	Access-accept

Name	Value	Type	Present in AAA message type
idlethreshold	integer	1	Access-accept, CoA
idle-timeout	integer	1	Access-accept, CoA
idle-timeout-direction	string	1	Access-accept, CoA
if-handle	integer	1	Accounting-request
inacl	string	1	Access-accept
intercept-id	integer	1	Access-accept
ip-addresses	string	1	Access-request, Accounting-request
ipv4-unnumbered **Note** This attribute-value pair (AVP) is preferred for BNG in Cisco IOS XR Software, and it is equivalent to the ip-unnumbered AVP in Cisco IOS Software.	string	1	Access-accept
ipv6_inacl	string	1	Access-accept, CoA
ipv6_outacl	string	1	Access-accept, CoA
ipv6-addr-pool	string	1	Access-accept
ipv6-dns-servers-addr	string	1	Access-accept
ipv6-enable	integer	1	Access-accept
ipv6-mtu	integer	1	Access-accept
ipv6-strict-rpf	integer	1	Access-accept
ipv6-unreachable	integer	1	Access-accept
l2tp-tunnel-password	string	1	Access-accept
ipv6 nd start-ra-on-ipv6-enable	Integer	1	Access-accept
login-ip-host	string	1	Accounting-request
maximum-interleaving-delay-downstream	integer	1	Access-request, Accounting-request
maximum-interleaving-delay-upstream	integer	1	Access-request, Accounting-request
maximum-data-rate-downstream	integer	1	Access-request, Accounting-request
maximum-data-rate-upstream	integer	1	Access-request, Accounting-request
md-dscp	integer	1	Access-accept

Name	Value	Type	Present in AAA message type
md-ip-addr	ipaddr	1	Access-accept
md-port	integer	1	Access-accept
minimum-data-rate-downstream	integer	1	Access-request, Accounting-request
minimum-data-rate-downstream-low-power	integer	1	Access-request, Accounting-request
minimum-data-rate-upstream	integer	1	Access-request, Accounting-request
minimum-data-rate-upstream-low-power	integer	1	Access-request, Accounting-request
outacl	string	1	Access-accept
parent-if-handle	integer	1	Access-request, Accounting-request
parent-session-id	string	1	Accounting-request
pppoe_session_id	integer	1	Accounting-request
primary-dns	ipaddr	1	Access-accept
qos-policy-in	string	1	Access-accept, CoA
qos-policy-out	string	1	Access-accept, CoA
redirect-vrf	string	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
sub-qos-policy-in	string	1	Access-accept
sub-qos-policy-out	string	1	Access-accept
Tunnel-Client-endpoint	ipaddr	1	Access-accept, Accounting-request
tunnel-id	string	1	Access-accept
tunnel-medium-type	string	1	Access-accept

Name	Value	Type	Present in AAA message type
Tunnel-Server-endpoint	ipaddr	1	Access-accept, Accounting-request
tunnel-tos-reflect	string	1	Access-accept
tunnel-tos-setting	integer	1	Access-accept
tunnel-type	string	1	Access-accept
username	string	1	Access-request, Accounting-request
vpdn-template	string	1	Access-accept
vpn-id	string	1	Access-accept
vpn-vrf	string	1	Access-accept
vrf-id	integer	1	Access-accept, Accounting-request
wins-server	ipaddr	1	Access-accept

Vendor-Specific Attributes for Account Operations

Table 18: Supported Vendor-Specific Attributes for Account Operations

RADIUS AVP	Value	Type	Action
subscriber:command=account-logon	string	1	account logon
subscriber:command=account-logoff	string	1	account logoff
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 19: Supported RADIUS ADSL Attributes

Name	Value	Type
Access-Loop-Encapsulation	binary	144
Actual-Interleaving-Delay-Downstream	integer	142
Actual-Interleaving-Delay-Upstream	integer	140
Actual-Data-Rate-Downstream	integer	130
Actual-Data-Rate-Upstream	integer	129

Name	Value	Type
Attainable-Data-Rate-Downstream	integer	134
Attainable-Data-Rate-Upstream	integer	133
Agent-Circuit-Id	string	1
IWF-Session	boolean social	254
Maximum-Interleaving-Delay-Downstream	integer	141
Maximum-Interleaving-Delay-Upstream	integer	139
Maximum-Data-Rate-Downstream	integer	136
Maximum-Data-Rate-Upstream	integer	135
Minimum-Data-Rate-Downstream	integer	132
Minimum-Data-Rate-Downstream-Low-Power	integer	138
Minimum-Data-Rate-Upstream	integer	131
Minimum-Data-Rate-Upstream-Low-Power	integer	137
Agent-Remote-Id	string	2

RADIUS ASCEND Attributes

Table 20: 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
Ascend-Multilink-Session-ID	integer	187
Ascend-Num-In-Multilink	integer	188

RADIUS Microsoft Attributes

Table 21: Supported RADIUS Microsoft Attributes

Name	Value	Type
MS-1st-NBNS-Server	ipv4addr	30
MS-2nd-NBNS-Server	ipv4addr	31

Name	Value	Type
MS-CHAP-ERROR	binary	2
MS-Primary-DNS	ipv4addr	28
MS-Secondary-DNS	ipv4addr	29

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 22: Supported Disconnect-Cause Attributes

Cause Code	Value	Description
0	No-Reason	No reason is given for the disconnect.
1	No-Disconnect	The event was not disconnected.
2	Unknown	Reason unknown.
3	Call-Disconnect	The call has been disconnected.
4	CLID-Authentication-Failure	Failure to authenticate number of the calling-party.
9	No-Modem-Available	A modem in not available to connect the call.
10	No-Carrier	No carrier detected. **Note** Codes 10, 11, and 12 can be sent if there is a disconnection during initial modem connection.
11	Lost-Carrier	Loss of carrier.
12	No-Detected-Result-Codes	Failure to detect modem result codes.

Cause Code	Value	Description
20	User-Ends-Session	User terminates a session. **Note** Codes 20, 22, 23, 24, 25, 26, 27, and 28 apply to EXEC sessions.
21	Idle-Timeout	Timeout waiting for user input. **Note** Codes 21, 100, 101, 102, and 120 apply to all session types.
22	Exit-Telnet-Session	Disconnect due to exiting Telnet session.
23	No-Remote-IP-Addr	Could not switch to SLIP/PPP; the remote end has no IP address.
24	Exit-Raw-TCP	Disconnect due to exiting raw TCP.
25	Password-Fail	Bad passwords.
26	Raw-TCP-Disabled	Raw TCP disabled.
27	Control-C-Detected	Control-C detected.
28	EXEC-Process-Destroyed	EXEC process destroyed.
29	Close-Virtual-Connection	User closes a virtual connection.
30	End-Virtual-Connection	Virtual connected has ended.
31	Exit-Rlogin	User exists Rlogin.
32	Invalid-Rlogin-Option	Invalid Rlogin option selected.
33	Insufficient-Resources	Insufficient resources.
40	Timeout-PPP-LCP	PPP LCP negotiation timed out. **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.
43	Failed-PPP-CHAP-Auth	PPP CHAP authentication failed.
44	Failed-PPP-Remote-Auth	PPP remote authentication failed.
45	PPP-Remote-Terminate	PPP received a Terminate Request from remote end.

Cause Code	Value	Description
46	PPP-Closed-Event	Upper layer requested that the session be closed.
47	NCP-Closed-PPP	PPP session closed because there were no NCPs open.
48	MP-Error-PPP	PPP session closed because of an MP error.
49	PPP-Maximum-Channels	PPP session closed because maximum channels were reached.
50	Tables-Full	Disconnect due to full terminal server tables.
51	Resources-Full	Disconnect due to full internal resources.
52	Invalid-IP-Address	IP address is not valid for Telnet host.
53	Bad-Hostname	Hostname cannot be validated.
54	Bad-Port	Port number is invalid or missing.
60	Reset-TCP	TCP connection has been reset. **Note** Codes 60 through 67 apply to Telnet or raw TCP sessions.
61	TCP-Connection-Refused	TCP connection has been refused by the host.
62	Timeout-TCP	TCP connection has timed out.
63	Foreign-Host-Close-TCP	TCP connection has been closed.
64	TCP-Network-Unreachable	TCP network is unreachable.
65	TCP-Host-Unreachable	TCP host is unreachable.
66	TCP-Network-Admin Unreachable	TCP network is unreachable for administrative reasons.
67	TCP-Port-Unreachable	TCP port in unreachable.
100	Session-Timeout	Session timed out.
101	Session-Failed-Security	Session failed for security reasons.
102	Session-End-Callback	Session terminated due to callback.
120	Invalid-Protocol	Call refused because the detected protocol is disabled.
150	RADIUS-Disconnect	Disconnected by RADIUS request.

Cause Code	Value	Description
151	Local-Admin-Disconnect	Administrative disconnect.
152	SNMP-Disconnect	Disconnected by SNMP request.
160	V110-Retries	Allowed V.110 retries have been exceeded.
170	PPP-Authentication-Timeout	PPP authentication timed out.
180	Local-Hangup	Disconnected by local hangup.
185	Remote-Hangup	Disconnected by remote end hangup.
190	T1-Quiesced	Disconnected because T1 line was quiesced.
195	Call-Duration	Disconnected because the maximum duration of the call was exceeded.
600	VPN-User-Disconnect	Call disconnected by client (through PPP). Code is sent if the LNS receives a PPP terminate request from the client.
601	VPN-Carrier-Loss	Loss of carrier. This can be the result of a physical line going dead. Code is sent when a client is unable to dial out using a dialer.
602	VPN-No-Resources	No resources available to handle the call. Code is sent when the client is unable to allocate memory (running low on memory).
603	VPN-Bad-Control-Packet	Bad L2TP or L2F control packets. This code is sent when an invalid control packet, such as missing mandatory Attribute-Value pairs (AVP), from the peer is received. When using L2TP, the code will be sent after six retransmits; when using L2F, the number of retransmits is user configurable. **Note** VPN-Tunnel-Shut will be sent if there are active sessions in the tunnel.

Cause Code	Value	Description
604	VPN-Admin-Disconnect	Administrative disconnect. This can be the result of a VPN soft shutdown, which is when a client reaches maximum session limit or exceeds maximum hopcount. Code is sent when a tunnel is brought down by issuing the clear vpdn tunnel command.
605	VPN-Tunnel-Shut	Tunnel teardown or tunnel setup has failed. Code is sent when there are active sessions in a tunnel and the tunnel goes down. **Note** This code is not sent when tunnel authentication fails.
606	VPN-Local-Disconnect	Call is disconnected by LNS PPP module. Code is sent when the LNS sends a PPP terminate request to the client. It indicates a normal PPP disconnection initiated by the LNS.
607	VPN-Session-Limit	VPN soft shutdown is enabled. Code is sent when a call has been refused due to any of the soft shutdown restrictions previously mentioned.
608	VPN-Call-Redirect	VPN call redirect is enabled.

APPENDIX C

Action Handlers

An action handler performs specific tasks in response to certain events. The following action handlers are currently supported in BNG.

Authorization Action Handler

The authorization action handler obtains authorization data for a specific subscriber identity from external AAA servers. The authorization action handler is an asynchronous function. It collects identity information from Subscriber Attribute Database (SADB) as well as the user credential data based on the identifier type specified in the CLI. This information along with method list name is sent to the AAA authorization coordinator. Once the AAA processing is done, the control is returned to the Policy Rule Engine (PRE) action handler to complete the event processing. The configuration example is as follows:

```
1 authorize aaa list <list-name> [identifier <identifier-type> | format <format_name>
password ['use-from-line'| <user-cfg-password>
```

Note Password is a mandatory, regardless of whether the user selects use-from-line or provides a specific value to use for authorization.

Authentication Action Handler

The authentication action handler gathers information like protocol type, service type, authentication type, user name, chap attributes, and user password and passes them to the AAA coordinator along with the AAA method list name. The authentication action handler is an asynchronous function. Once the AAA processing is done, the control is returned to the PRE action handler to complete the event processing. The configuration example is as follows:

```
1 authenticate aaa list <list-name>
```

Disconnect Action Handler

The disconnect action handler is called to disconnect a subscriber. For a subscriber disconnect, the PRE informs the Policy Plane Session Manager (PPSM) to notify all clients about the subscriber disconnect. The PPSM reports back to the PRE to complete the disconnection. The PRE puts the subscriber in the disconnect state. The PRE also cleans-up the record history data that stores policy execution history and the control block containing the subscriber label. When PRE processing is done, control is returned to the PPSM for further processing.

Activate Action Handler

The activate action handler enables local dynamic templates or remote AAA services on the subscriber's configuration. The results of this action are either immediate or asynchronous. The PRE gathers information like the AAA method list name, template type, and template name and sends to the SVM for processing. The SVM returns the control after completing template processing, and the PRE resumes processing the action list from the place it had stopped. The configuration example is as follows:

```
1 activate dynamic-template <template-name> [aaa list <list-name>]
```

Deactivate Action Handler

The deactivate action handler disables local dynamic templates or remote AAA services from the subscriber's configuration. The result of this action is asynchronous. The PRE collects information like AAA list, template type, and template name and sends to the SVM. to request it to not apply the service. The AAA list is used to derive a key used in SVM. SVM returns control after completing template processing, and the PRE restarts processing the action list from where it had stopped. The configuration example is as follows:

```
1 deactivate dynamic-template <template-name> [aaa list <list-name>]
```

Set Timer and Stop Timer Action Handlers

The set timer action handler sets an active named timer for a defined time period on the subscriber session. The stop timer stops an active named timer on the subscriber session. Enabling the set timer action handler allows the service provider to have one or more timed-policy-expiry events to be triggered on a subscriber. This in turn provides better subscriber management over the subscriber life cycle. These action handlers provide functions like scheduled validation of subscriber state status (checking if the subscriber is authenticated or unauthenticated) and periodically changing subscriber policy (such as forcing re-authentication on a daily or hourly basis).

Note An action with a timer value of 0, triggers the action immediately.

There are two methods to stop an active timer:

- Allow the timer to expire.
- Stop the active running timer using the stop-timer action command.

APPENDIX D

BNG Use Cases and Sample Configurations

This appendix describes the various BNG use cases and sample configurations:

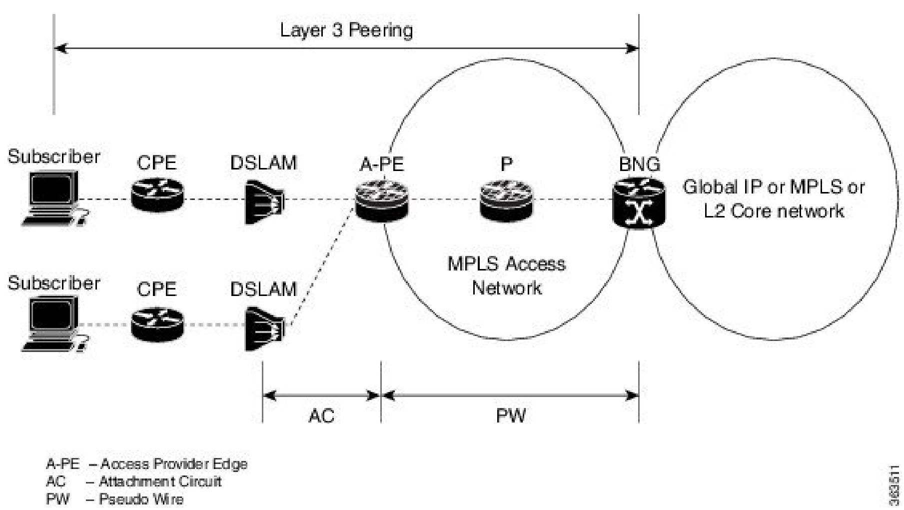
BNG over Pseudowire Headend

Sample Topology for BNG over Pseudowire Headend

For an overview of BNG over Pseudowire Headend, see BNG over Pseudowire Headend , on page 170.

This figure shows a sample topology for BNG over Pseudowire Headend:

Figure 26: Sample Topology for BNG over Pseudowire Headend



Deployment Models for Subscribers on Pseudowire Headend

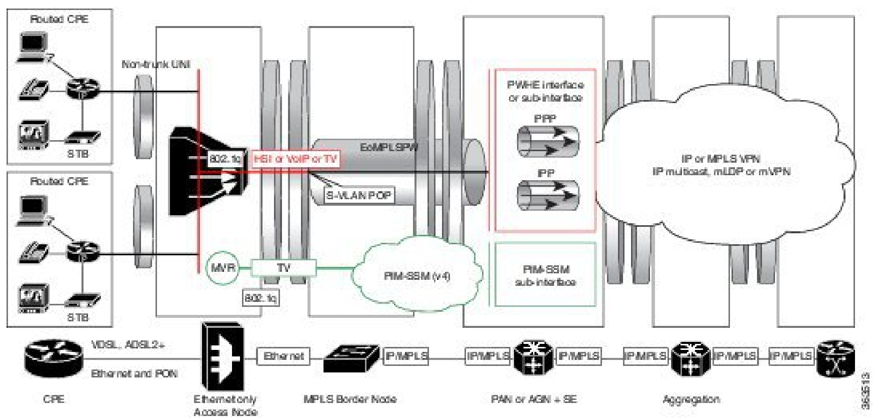
Residential Subscribers on Pseudowire Headend

The deployment models available for residential subscribers on PWHE are:

N:1 model

This figure shows the n:1 deployment model for residential subscribers on PWHE:

Figure 27: N:1 deployment model for residential subscribers on PWHE

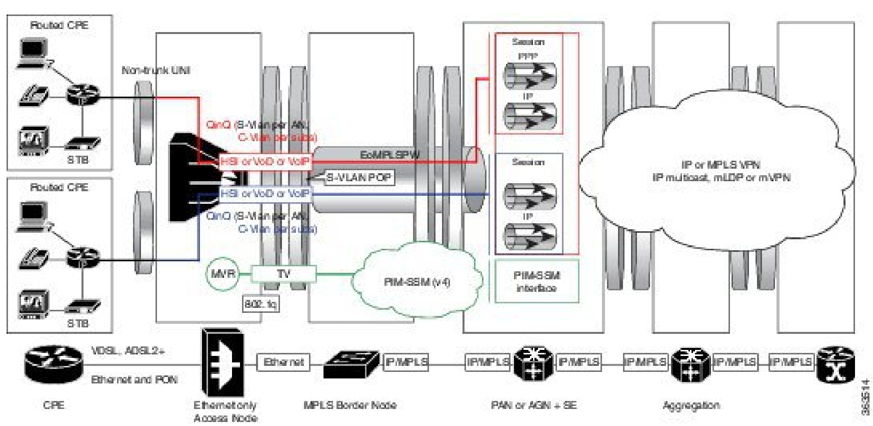


This model does not have subscriber VLANs. All subscribers connected to the DSLAM are aggregated into an S-VLAN and sent to the BNG, over a pseudowire. In most cases, there is only one pseudowire for each DSLAM in this deployment model. In this model, the pseudowire can be negotiated for VC type 4 and the subscriber can be terminated on the PWHE main interface. The pseudowire can also be negotiated for VC type 5 and be matched with the PWHE sub-interface that is configured for the S-VLAN (if VLAN is retained in the pseudowire).

1:1 model

This figure shows the 1:1 deployment model for residential subscribers on PWHE:

Figure 28: 1:1 deployment model for residential subscribers on PWHE



In this model, the subscriber traffic comes in VLANs to the DSLAM and one pseudowire is created per DSLAM. Here, the pseudowire is negotiated for VC type 5, and therefore, the S-VLAN is not retained in the pseudowire. The subscriber VLANs can be matched with the PWHE sub-interface configuration. There cannot be a matching sub-interface for each subscriber VLAN. As a result, ambiguous VLANs must be enabled on the PWHE sub-interfaces to accommodate multiple unique subscriber VLANs.

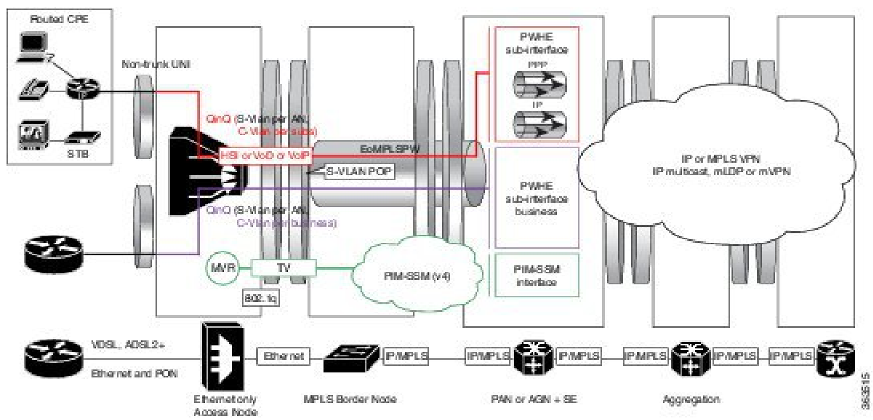
Residential and Business Subscribers on Pseudowire Headend

The deployment models available for residential and business subscribers on PWHE are:

Model 1

This figure shows the deployment model 1 for residential and business subscribers on PWHE:

Figure 29: Deployment Model 1 for Residential and Business Subscribers on PWHE

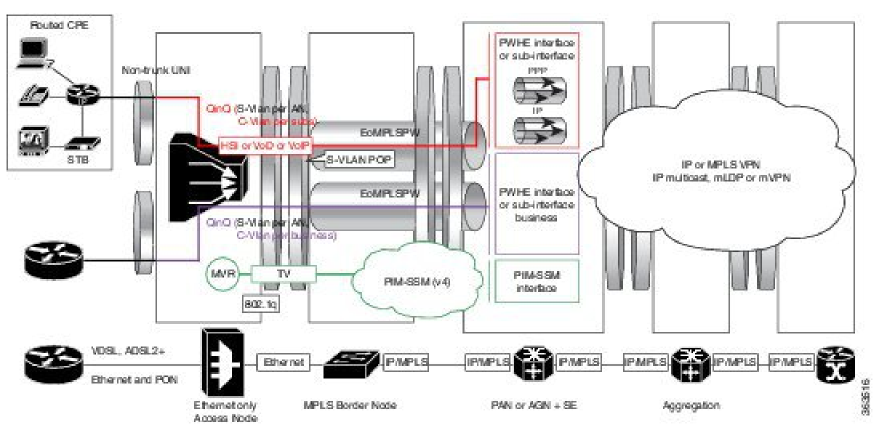


In this model, all services from the access network are enabled on different sub-interfaces on the same pseudowire. The PW is negotiated for VC type 5. This solution model provides up to service level aggregation; an aggregate shaper may not be applied on the main interface.

Model 2

This figure shows the deployment model 2 for residential and business subscribers on PWHE:

Figure 30: Deployment Model 2 for Residential and Business Subscribers on PWHE



In this model, all services from the access network are enabled on PWHE sub-interfaces configured on different pseudowires. The PW is negotiated for VC type 5. An aggregate shaper can also be applied on both the PWHE interfaces.

Configuration and Verification of BNG over Pseudowire Headend

Configuration Commands for BNG over Pseudowire Headend

These are some of the common commands to be used to configure BNG over Pseudowire Headend:

Table 23: Configuration Commands for BNG over Pseudowire Headend

Command	Purpose
pw-class *class-name*	Configures the pseudowire class template name to use for the pseudowire.
encapsulation mpls	Configures the pseudowire encapsulation to MPLS.
protocol ldp	Sets pseudowire signaling protocol to LDP.
xconnect group *group-name*	Configures a cross-connect group name using a free-format 32-character string.
l2overhead *bytes*	Sets layer 2 overhead size.
generic-interface-list *bytes*	Configures a generic interface list.
attach generic-interface-list *interface_list_name*	Attaches the generic interface list to the PW-Ether or PW-IW interface.
encapsulation dot1q *vlan-id*	Assigns the matching VLAN-Id and Ethertype to the interface.
QoS Commands	
service-policy output *policy-name* [**subscriber-parent resource-id** *value*]	Configures egress SVLAN policy on PW-Ether sub interface.
service-policy output *policy-name* [**shared-policy-instance** *instance-name*]	Configures egress policy (with or without shared-policy-instance) on PWHE subscriber interface.

BNG Use Cases and Sample Configurations

Note For more information about the PWHE feature and the related configuration procedures in Cisco ASR9K router, see the *Implementing Multipoint Layer 2 Services* chapter in the *L2VPN and Ethernet Services Configuration Guide for Cisco ASR 9000 Series Routers*. For complete command reference of the PWHE-specific commands in Cisco ASR9K router, see the *VPN and Ethernet Services Command Reference for Cisco ASR 9000 Series Routers*.

For more information about QoS features and the related configuration in Cisco ASR9K router, see the *Modular QoS Configuration Guide for Cisco ASR 9000 Series Routers*. For complete command reference of the QoS-specific commands in Cisco ASR9K router, see the *Modular Quality of Service Command Reference for Cisco ASR 9000 Series Routers*.

Verification Commands for BNG over Pseudowire Headend

This table lists the verification commands for BNG over Pseudowire Headend.

Command	Purpose
show run lvpn	Displays the running configuration of L2VPN.
show run interface PW-Ether *interface-name*	Displays the running configuration of pw-ether interface.
show run mpls ldp	Displays the running configuration of MPLS ldp.
show run generic-interface-list	Displays the running configuration of generic-interface-list.
show l2vpn xconnect detail	Displays the configuration details of L2VPN cross-connect.
show l2vpn xconnect detail | include packet	Displays the configuration details of L2VPN cross-connect with lines that match *packet*.
show controller np counters all	Displays the counter statistics of network processors.
BNG-specific commands:	
show subscriber session all summary	Displays the summary of subscriber session information.
show subscriber manager disconnect history	Displays the disconnect history of subscriber manager.
show tech-support subscriber [ipoe | memory | pta]	Collects the output of relevant BNG subscriber related commands, and saves it to the local disk.
QoS commands:	
show policy-map interface	Displays the policy configuration information for all classes configured for all service policies on the specified interface.
show policy-map shared-policy-instance	Displays the statistics for all details of the shared policy instance.

Sample Configurations for BNG over Pseudowire Headend

This section provides the sample configurations for BNG over Pseudowire Headend (without QoS).

- PWHE Configuration

```
//l2vpn pw-class
l2vpn
 pw-class deep
  encapsulation mpls
   protocol ldp
   control-word
   transport-mode vlan
!
!
!

//l2vpn xconnect group
l2vpn
 xconnect group xc1
  p2p 101
   interface PW-Ether101
   neighbor 3.3.3.3 pw-id 2300
    pw-class deep

//Generic interface list configuration
generic-interface-list double1
 interface GigabitEthernet0/3/0/1
 interface Bundle-Ether 101
!

//pw-ether interface configuration
interface pw-ether101
 l2overhead 64
 attach generic-interface-list double1
mac-address <mac-address>
!

interface pw-ether 101.1
  encapsulation dot1q 10
 ipv6 address 1001::1/64
 ipv4 address 162.162.1.2 255.255.255.0
!
```

- Subscriber Configuration on PWHE access-interface

```
//IPoE
interface PW-Ether1.1
 ipv4 unnumbered Loopback200
  service-policy type control subscriber ISN_CNTRL_1
  ipsubscriber ipv4 l2-connected
  initiator dhcp
  initiator unclassified-source
 encapsulation ambiguous dot1q 73 second-dot1q any
!

//PPPoE
interface PW-Ether1.4
ipv6 enable
pppoe enable
```

```
service-policy type control subscriber pppoe_pxy
encapsulation dot1q 104
!
```

This section provides the sample configurations for BNG over Pseudowire Headend (with QoS).

- Egress SVLAN policy configuration on PW-Ether sub interface:

```
interface pw-ether 2.1
 ipv4 address 11.11.11.11 255.255.255.0
 encapsulation dot1q 100
 service-policy output policy1 subscriber-parent
!
```

- Egress policy (with or without shared-policy-instance) on PWHE subscriber interface:

```
interface pw-ether 2.1
 service-policy output policy1 shared-policy-instance
!
```

- Policy application on PWHE subscriber interface, with service accounting enabled:

```
dynamic-template
type ppp ppp1
ppp ipcp peer-address pool ppp_pool
ipv4 unnumbered Loopback10
!
type service S1
service-policy output test acct-stats
accounting aaa list default type service
!
!
```

- Policy application on PWHE subscriber through Radius CoA (pQoS):

```
qos-policy-{in | out}={add-class | remove-class} (sub,<parent-class,
child-class>,<action-list>)
2 - level policy-map definition
Each vsa defines one class and its actions

CoA / Access-Accept {
qos-policy-out=add-class(sub, (class-default), shape(2000))
qos-policy-out=add-class(sub, (class-default, data), shape(500), bw-rpct(25))
qos-policy-out=add-class(sub, (class-default, class-default), queue-limit(20000))
}

policy-map type qos __policy1_out
class class-default
shape average 2000 kbps
service-policy child1
!
end-policy-map

policy-map type qos __policy1_child1
class data
shape average 500 kbps
bandwidth remaining percent 25
```

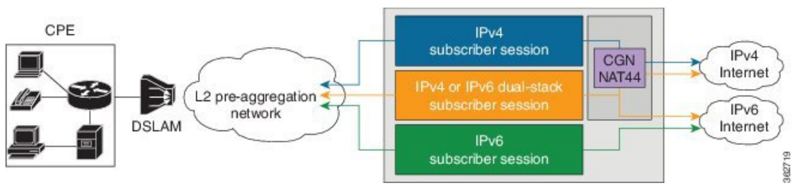
```
!
class class-default
queue-limit 20000 packets
!
end-policy-map
```

Dual-Stack Subscriber Sessions

The BNG supports dual-stack for subscriber sessions, whereby an IPv4 address and an IPv6 address can co-exist for the same subscriber.

The figure below shows a deployment model of dual-stack subscriber sessions.

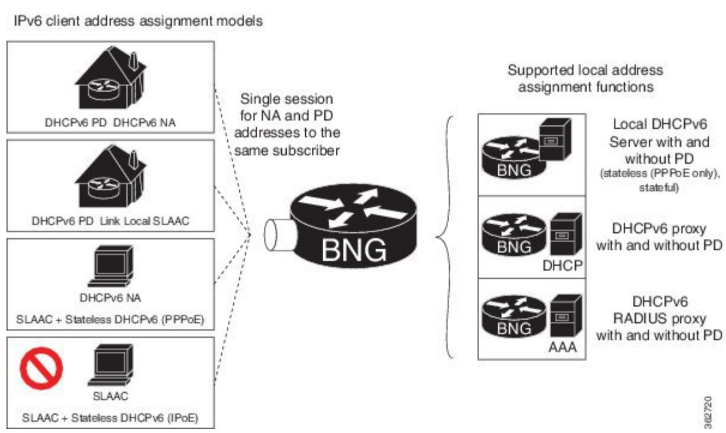
Figure 31: Deployment Model of Dual-Stack Subscriber Sessions



IP Address Assignment for Clients

The following figure shows various IP address assignment options available for IPv6 clients, and the supported local address assignment functions.

Figure 32: IPv6 Client Address Assignment Models



The **framed-ipv6-address** RADIUS attribute can also be used to provide an IP address from the RADIUS server to the subscriber. This address is then advertised through a Stateless Address Auto Configuration - Neighbor Advertisement or Neighbor Discovery (SLAAC - NA or ND) message for both PPPoE and IPoE sessions.

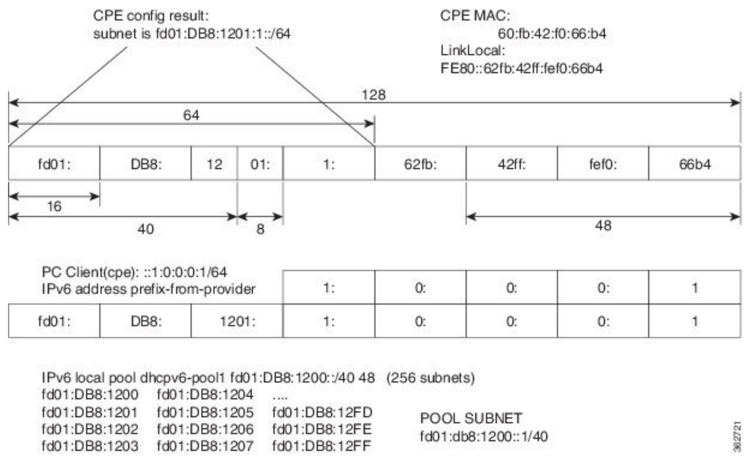
If DHCPv6 is not used for the IPoE sessions, an additional Vendor-Specific Attribute **ipv6:ipv6-default-gateway** is used to specify the default router.

Sample IPv6 Addressing and Configurations

IPv6 Address Mapping

The following figure shows the sample IPv6 address mapping with prefix-delegation in place, for the dual-stack subscriber. The respective sample CPE configurations and the sample DHCPv6 Server configurations are discussed in subsequent sections.

Figure 33: Sample IPv6 Address Mapping for Dual-Stack Subscriber



CPE Configurations

Sample Configuration for the Client Side of the CPE

This section provides the sample configurations for the client side of the Customer Premises Equipment (CPE).

```
interface GigabitEthernet0/2
description to switch fa0/15
ip address 192.168.1.1 255.255.255.0
no ip unreachables
ip nat inside
ip virtual-reassembly
duplex full
speed 100
media-type rj45
negotiation auto
ipv6 address prefix-from-provider ::1:0:0:0:1/64
ipv6 enable
```

Sample Configuration for the WAN Side of the CPE

This section provides the sample configurations for the WAN side of the Customer Premises Equipment (CPE).

```
interface FastEthernet2/0.50
encapsulation dot1Q 50
ipv6 address autoconfig default
ipv6 enable
```

```
ipv6 dhcp client pd prefix-from-provider
```

DHCPv6 Server Configuration

Sample Configuration for the DHCPv6 Server

This section gives the sample configurations for the DHCPv6 Server.

```
ipv6 unicast-routing
ipv6 dhcp pool dhcpv6
prefix-delegation pool dhcpv6-pool1 lifetime 6000 2000
ipv6 route 2001:60:45:28::/64 2005::1
ipv6 route 2001:DB8:1200::/40 2005::1
ipv6 route 200B::/64 2005::1
ipv6 route 2600:80A::9/128 4000::1
ipv6 local pool dhcpv6-pool1 2001:DB8:1200::/40 48
```

Note BNG supports only a single IA-NA and IA-PD for the subscribers. Therefore, if the ASR9K is configured as a DHCP server, and if the BNG subscriber sends a DHCPv6 SOLICIT message with more than one IA-NA and IA-PD, then the DHCP ADVERTISEMENT response from the ASR9K fails. And, the subscriber will not get the IPv6 address in such scenarios.

Operation and Call Flow of Dual-Stack Sessions

The ASR9K router considers the IPv4 and IPv6 stacks as a single subscriber. Therefore, only a single Access Request message and a single accounting record are generated for both the stacks. However, in scenarios such as the one where an accounting request is generated, the two stacks are considered as being two separate entities.

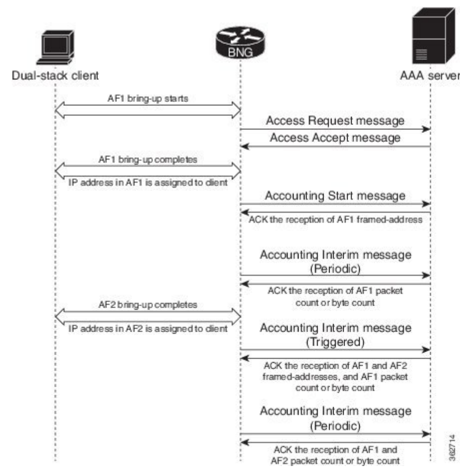
Note

- When the first address-family (AF) comes up, the Access Request message that is generated must contain, for the session, information about both the IPv4 and the IPv6. A second request is not generated for the other AF.
- When the first AF comes up, the BNG router generates an Accounting Start message and sends it to the AAA server. The BNG waits for a pre-determined period of time and generates a single accounting start record for both address-families. As another option, an interim accounting record is triggered by the BNG when the second AF comes up.

Generic Call Flow of Dual-Stack Session

The figure below shows the generic call flow of dual-stack session. The interactions with other servers, such as the DHCP server, are not displayed in this figure.

Figure 34: Generic Call Flow of Dual-Stack Session



The details of the call flow between the BNG router and the AAA server are listed here:

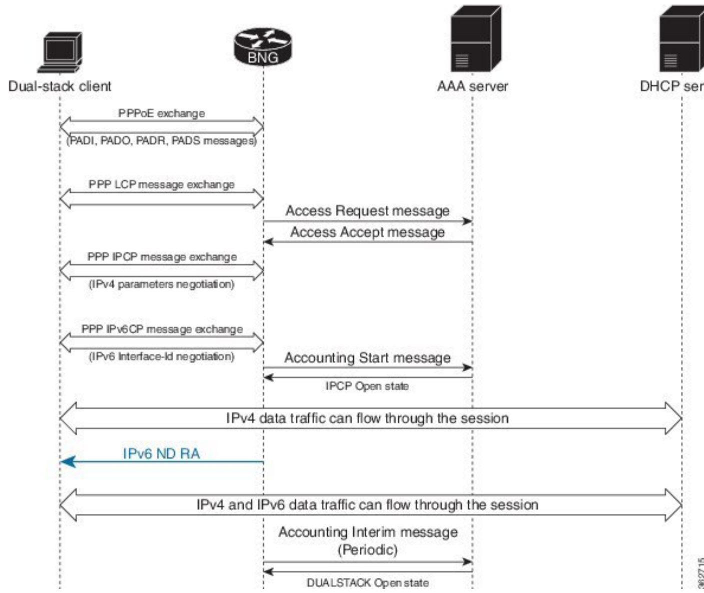
- A single authentication process for the first and the second address-family (AF1 and AF2) is triggered when the first AF1 comes up.
- A single Accounting Start message is triggered when the AF1 is set up. The framed-address for the AF1 that is set up, is sent from the AAA server back to the BNG router.
- The statistics for the AF that is currently set up (AF1 in this case) is sent through periodic Accounting Interim messages.
- The AF2 is set up next, and the statistics for the AF2 is sent through triggered Accounting Interim messages.
- The statistics for each AF and the aggregated statistics for both the address-families that are set up are sent by periodic Accounting Interim messages.

Detailed Call Flows - PPPoE Dual-Stack

Scenario 1: SLAAC-Based Address Assignment

The figure below shows the detailed call flow of PPPoE dual-stack, where the address assignment is SLAAC-based.

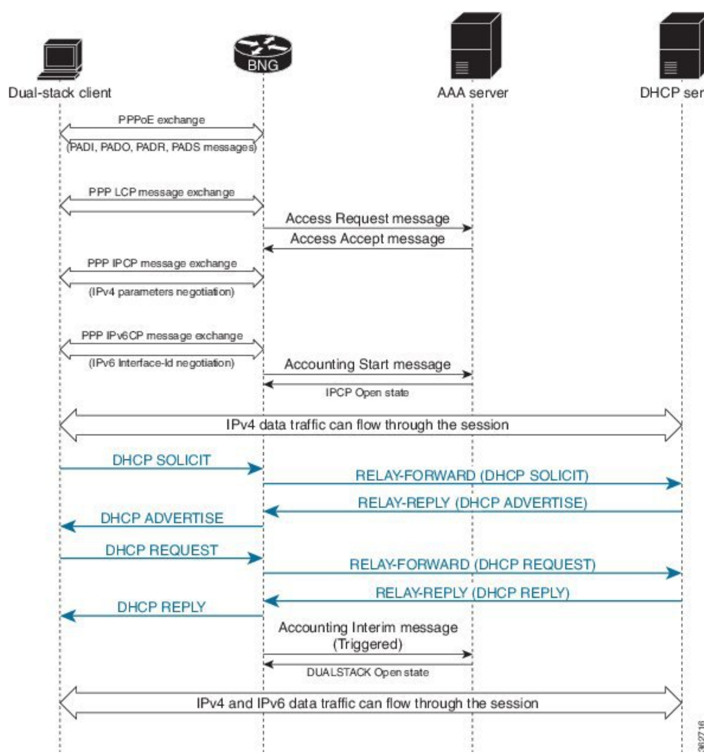
Figure 35: PPPoE Dual-Stack - SLAAC-Based Address Assignment



Scenario 2: DHCPv6-Based Address Assignment

The figure below shows the detailed call flow of PPPoE dual-stack, where the address assignment is DHCPv6-based.

Figure 36: Call Flow of PPPoE Dual-Stack - DHCPv6-Based Address Assignment

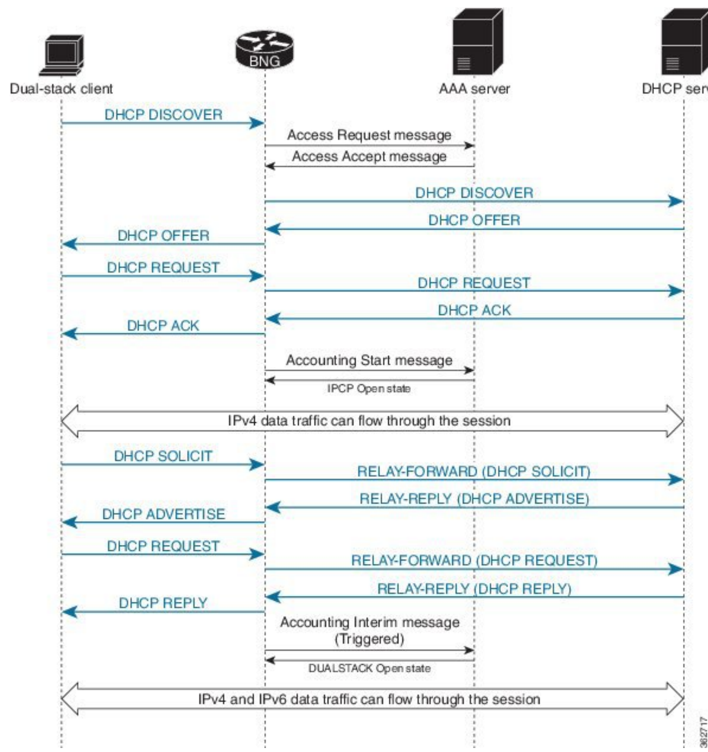


Detailed Call Flows - IPoE Dual-Stack

Scenario 1 - IPv4 Address-Family Starts First

The figure below shows the detailed call flow of IPoE dual-stack, where the IPv4 address-family (AF) starts first.

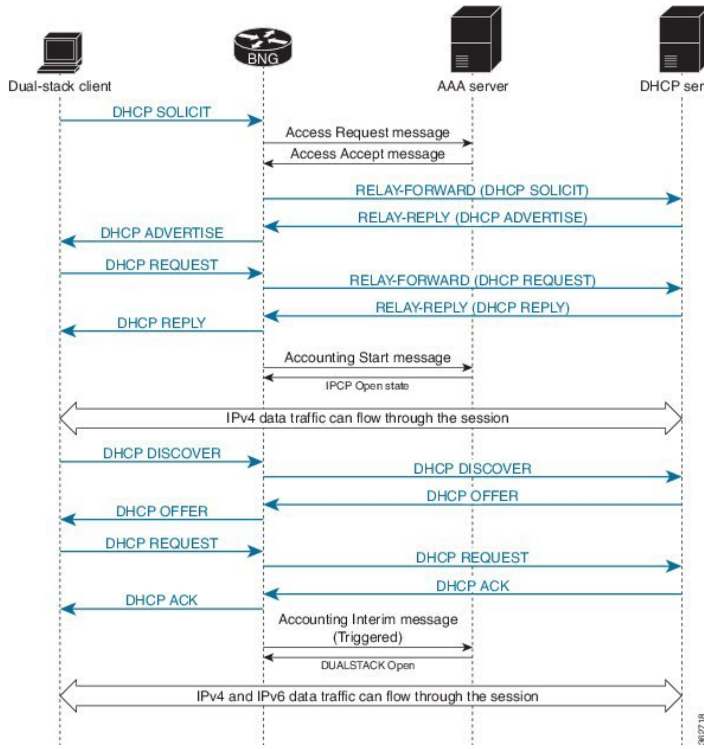
Figure 37: Call Flow of IPoE Dual-Stack - IPv4 Address-Family Starts First



Scenario 2 - IPv6 Address-Family Starts First

The figure below shows the detailed call flow of IPoE dual-stack, where the IPv6 address-family (AF) starts first.

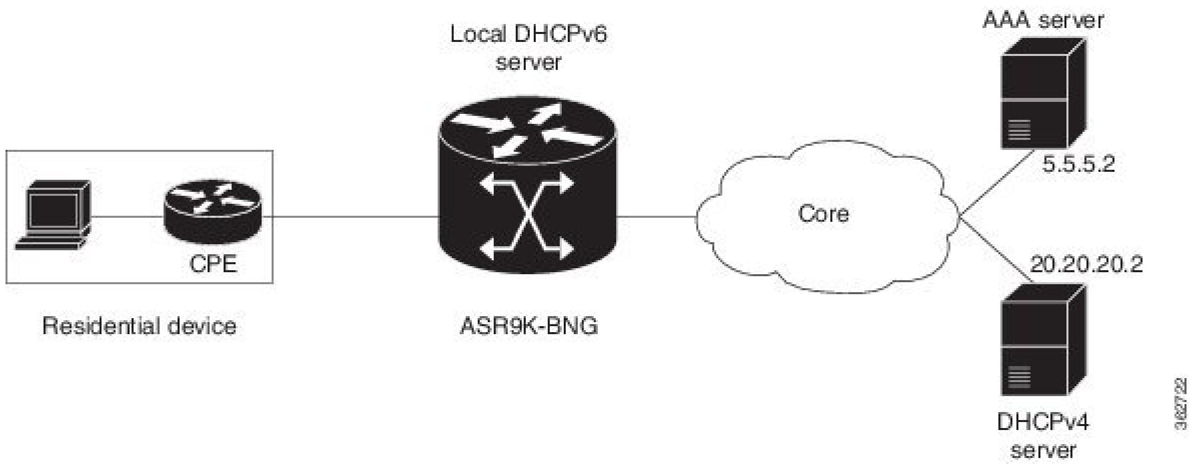
Figure 38: Call Flow of IPoE Dual-Stack - IPv6 Address-Family Starts First



Sample Topology for Dual-Stack

The figure below shows a sample topology for the dual-stack.

Figure 39: Sample topology for Dual-Stack



Configuration Examples for Dual-Stack

This section provides configuration examples for a dual-stack.

```
hostname bng
logging console debugging
```

The RADIUS server is configured with the server listening on the IP address 5.5.5.2 with **auth-port** on 1645 and **accounting-port** on 1646.

```
radius-server host 5.5.5.2 auth-port 1645 acct-port 1646
key 7 010107000A5955
!
```

The CoA server or policy-server with IP address 5.5.5.2 is configured.

```
aaa server radius dynamic-author
client 5.5.5.2 vrf default server-key 7 03165A0F575D72
!
aaa group server radius RADIUS
server 5.5.5.2 auth-port 1645 acct-port 1646
!
aaa accounting   service default group radius
aaa accounting   subscriber default group radius
aaa authorization subscriber default group radius
aaa authentication subscriber default group radius
!
```

The DHCPv6 address pool is defined locally within the BNG router and the local pool is used for IPv6 address assignment to the IPv6 BNG clients.

```
pool vrf default   ipv6 ipv6_address_pool
address-range 2001::2 2001::7dff
!
```

The DHCPv4 server with IP address 20.20.20.2 is deployed externally and this IPv4 address must be reachable from the BNG router. The routing protocols must take care of the reachability of the IP address 20.20.20.2 from the BNG router. The DHCPv4 proxy is configured, thus:

```
dhcp ipv4
profile IPoEv4 proxy
helper-address vrf default 20.20.20.2 giaddr 10.10.10.1
```

The DHCPv4 proxy is enabled on the bundle sub-interface.

```
interface Bundle-Ether1.10 proxy profile IPoEv4
!
```

The DHCPv6 server is configured and the previously-configured DHCPv6 address pool is referred within the DHCPv6 server configuration. The DHCPv6 profile along with the address-pool is configured, thus:

```
dhcp ipv6
   profile IPoEv6 server
   address-pool ipv6_address_pool
!
```

The DHCPv6 address pool is referred on the bundle sub-interface.

```
interface Bundle-Ether1.10 server profile IPoEv6
!
interface Bundle-Ether1
bundle maximum-active links 1
!
```

The bundle sub-interface with the dot1q encapsulation is configured with a single tag. The subscriber traffic from the CPE should come with the single dot1q tag and this VLAN tag must match the VLAN-ID 10 configured under the bundle sub-interface. In Dual-Stack IPoE configuration, the **initiator dhcp** command is configured under the IPv4 or IPv6 l2-connected configuration mode. The name of the policy-map type control is referred with the service-policy.

```
interface Bundle-Ether1.10
ipv4 point-to-point
ipv4 unnumbered Loopback1
ipv6 enable
service-policy type control subscriber pm-src-mac
encapsulation dot1q 10
ipsubscriber ipv4 l2-connected
initiator dhcp
!

ipsubscriber ipv6 l2-connected
initiator dhcp
!
!
```

The IPv4 address 10.10.10.1 is the default-gateway IP address for the pool of IPv4 addresses allocated to the dual-stack BNG clients.

```
interface Loopback1
ipv4 address 10.10.10.1 255.255.255.0
ipv6 enable
!
```

The physical interface GigabitEthernet0/0/0/0 is configured as the bundle interface.

```
interface GigabitEthernet0/0/0/0
bundle id 1 mode on
negotiation auto
transceiver permit pid all
!
```

The dual-stack dynamic-template is configured for the dual-stack initiation. The IPv6 enable, IPv4 unnumbered address and IPv4 urpf are configured under the dual-stack template.

```
dynamic-template
   type ipsubscriber Dual_stack_IPoE
     accounting aaa list default type session periodic-interval 5
     ipv4 verify unicast source reachable-via rx
     ipv4 unnumbered Loopback1
     ipv6 enable
!
```

```
!
```

The class-map is configured for the dual-stack scenario in order to match the DHCPv6 - SOLICIT and DHCPv4 - DISCOVER messages as the first-sign-of-life (FSOL) packets.

```
class-map type control subscriber match-any dual_stack_class_map
   match protocol dhcpv4 dhcpv6
   end-class-map
!
```

The **dual_stack_class_map** class-map is referred within the policy-map. The event session-start is matched based on the DHCPv4 or DHCPv6 FSOL and the **Dual_stack_IPoE** dynamic-template is activated. The subscriber Mac-Address is used for subscriber identification, and this address is authorized with the AAA server.

```
policy-map type control subscriber pm-src-mac
event session-start match-all
   class type control subscriber dual_stack_class_map do-all
     1 activate dynamic-template Dual_stack_IPoE
     2 authorize aaa list default identifier source-address-mac password cisco
!
!
end-policy-map
!
end
```

Verification Steps for Dual-Stack

This section provides the list of commands that can be used for verifying dual-stack configuration. For details of these commands, see *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*.

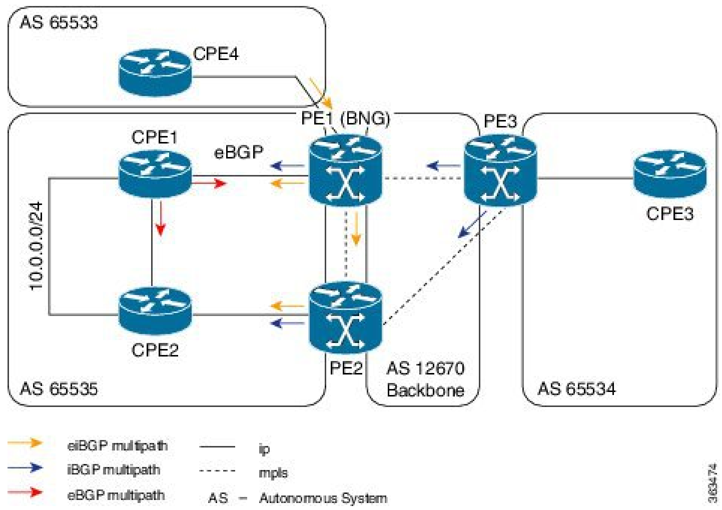
Command	Purpose
show subscriber session all	Displays active IPv4 or IPv6 client sessions.
show subscriber session all detail	Displays details of active IPv4 or IPv6 client sessions.
show dhcp ipv4 proxy binding	Displays IPoEv4 clients created. It displays the **ip-address**, **mac-address**, the interface on which the IPoEv4 clients are created, the **vrf-name**, and so on.
show dhcp ipv4 proxy binding detail	Displays details of IPoEv4 clients created.
show dhcp ipv6 server binding	Displays IPv6 address allocated from the DHCPv6 local pool.

eBGP over PPPoE

Sample Topology for eBGP over PPPoE

This figure shows a sample topology for eBGP over PPPoE:

Figure 40: Sample topology for eBGP over PPPoE



All Provide Edge (PE) routers shown in the figure, are in the service provider Autonomous System (AS), representing the core of the network. The CPE1 and CPE2 are in customer AS, peering with the PEs (PE1 and PE2 respectively) as eBGP neighbors. A statically-configured loopback address on CPE and PE, is used for BGP peering. There are networks behind the CPE, and the CPE advertises respective prefixes to the PE routers through eBGP.

PE1 and PE2 that are configured as BNG, provide reachability to the same CPE. Along with site bring-up, CPE1 and CPE2 tries to establish subscriber sessions with PE1 and PE2 respectively. These can be PPPoE PTA sessions. As part of authentication, BNG receives RADIUS attributes (through an Access-accept message) and brings-up subscribers on the respective customer VRF. The Access-accept message also contains a Framed-Route attribute that sets up a route to the CPE loopback through the subscriber interface.

When the subscriber session is up, the BGP on the CPE and the PE discover each other as neighbors and start exchanging prefixes. Based on the BGP configuration on the BNG, a label is allocated for each prefix that is advertised by the CPE. As CPE1 and CPE2 are advertising routes for the same network (For example, 10.0.0.0/24), both PE1 and PE2 allocate a label for the same prefix and distribute it to each other, and to PE3. All PEs have eiBGP multi-path enabled through the configuration, and they keep multiple paths active in the FIB chain. For example, PE3 can reach 10.0.0.0/24 through PE1 or PE2. Similarly, PE1 can reach the network through CPE1 or through PE2-CPE1. The traffic is equally distributed across all available paths. The PE1 and PE2 must be configured such that when the same prefix is advertised, a loop does not occur due to multiple path creation. In that case, only PE1 accepts the route, and PE2 must be configured to reject multiple paths.

Configuration and Verification of eBGP over PPPoE

Configuration Commands

These are some of the common BGP and MPLS commands used to configure eBGP over PPPoE:

Table 24: Configuration Commands for eBGP over PPPoE

Command	Purpose
maximum-paths eibgp *num_path*	Configures the maximum number of eibgp multi-paths allowed under a VRF.

Command	Purpose
maximum-paths ibgp *num_path*	Configures the maximum number of ibgp paths allowed under a VRF.
ebgp-multihop *num_hop*	Sets the number of hops by which the ebgp neighbor is from the PE.
label mode per-prefix	Sets the label mode as per-prefix, for prefixes learnt over eBGP.
cef load-balancing	Configures the load-balancing functionality at each node.

For details on BGP configurations, see *Implementing BGP* chapter in *Routing Configuration Guide for Cisco ASR 9000 Series Routers*. For complete command reference of BGP Commands, see *Border Gateway Protocol Commands* chapter in *Routing Command Reference for Cisco ASR 9000 Series Routers*.

For details on MPLS configurations, see *Implementing MPLS Label Distribution Protocol* chapter in *MPLS Configuration Guide for Cisco ASR 9000 Series Routers*. For complete command reference of MPLS Commands, see *MPLS Command Reference for Cisco ASR 9000 Series Routers*.

Troubleshooting Steps for eBGP over PPPoE

As part of troubleshooting eBGP over PPPoE, verify these:

- Ensure that the **maximum-paths eibgp** command is configured for eiBGP multi-path.
- If iBGP multi-path is failing, verify the metric or cost. If they are unequal, configure the **maximum-paths ibgp** *num_path* **unequal-cost** command for iBGP.
- If traffic is not flowing on all paths, verify **cef load-balancing**. It must have at least L3 hash, and traffic must be sent with different source and destination IP addresses.
- If multi-path is not working, perform these:
    - Verify whether both Routing Information Base (RIB) and Cisco Express Forwarding (CEF) have two paths.
    - Verify BGP neighbors, whether routers do get exchanged.
    - Verify whether eBGP neighbor is reachable through static route, and ensure that **ebgp multi-hop** is configured in BGP configuration.

Verification Commands for eBGP over PPPoE

These show commands are used to verify the eBGP over PPPoE configurations:

Table 25: Verification Commands for eBGP over PPPoE

Command	Purpose
show route vrf *vrf_name network_IP* detail	Displays the current contents of the RIB. This can be used to verify whether both RIB and CEF have two paths, when multi-path is not working.

Command	Purpose
show cef vrf *vrf_name* detail	Displays the CEF-related information for a VRF.
show bgp vpnv4 unicast *network_IP*	Displays entries related to VPNv4 unicast address families in BGP routing table.
show bgp vpnv4 unicast neighbors	Displays detailed information on TCP and BGP neighbor connections.
show bgp neighbors	Displays information about BGP neighbors, including configuration inherited from neighbor groups, session groups, and address family groups.

Sample Configurations for eBGP over PPPoE

This section provides some sample configurations for eBGP over PPPoE:

- PE1 Configuration

```
//VRF Configuration

vrf CPE_1_VRF_1
address-family ipv4 unicast
  import route-target
   200:1
   200:3
   200:4
  !
  export route-target
   200:1
  !
!
!

vrf CPE_4_VRF_1
address-family ipv4 unicast
  import route-target
   200:1
   200:3
   200:4
  !
  export route-target
   200:4
  !
!
!

//BGP Configuration

route-policy EBGP_ROUTE_POLICY
pass
end-policy

router bgp 200
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
```

BNG Use Cases and Sample Configurations

```
 neighbor 65.0.0.2 --->PE2
  remote-as 200
  update-source Loopback0
  address-family vpnv4 unicast
   !
 !
 neighbor 65.0.0.3 --->PE3
  remote-as 200
  update-source Loopback0
  address-family vpnv4 unicast
   !
 !

//maximum-paths and per-prefix label mode configurations

vrf CPE_1_VRF_1
  rd 65001:1
  address-family ipv4 unicast
   maximum-paths eibgp 8
   label mode per-prefix
  !
  neighbor 101.0.0.1 --->CPE1
   remote-as 65535
   ebgp-multihop 5
   update-source Loopback1
   address-family ipv4 unicast
    route-policy EBGP_ROUTE_POLICY in
    route-policy EBGP_ROUTE_POLICY out
   !
  !
 !

vrf CPE_4_VRF_1
  rd 65004:1
  address-family ipv4 unicast
   maximum-paths eibgp 8
   label mode per-prefix
  !
  neighbor 104.0.0.1 --->CPE4
   remote-as 65533
   update-source Loopback5001
   address-family ipv4 unicast
    route-policy EBGP_ROUTE_POLICY in
    route-policy EBGP_ROUTE_POLICY out
   !
  !
 !

//RADIUS Configuration

DEFAULT Cleartext-Password :=cisco, Nas-Port-Id == "0/0/50/2"
        Framed-Protocol = PPP,
        Framed-IP-Address = 11.11.0.1,
        Framed-Route = "101.0.0.1 255.255.255.255 0.0.0.0 6 tag 7",
        Service-Type = Framed-User,
        Cisco-Avpair += "ipv4:ipv4-unnumbered=Loopback1",
        Cisco-avpair += "subscriber:vrf-id=CPE_1_VRF_1",

//MPLS Configuration

mpls ldp
router-id 65.0.0.1 --->Local IP
interface GigabitEthernet0/0/1/9
!
```

```
interface GigabitEthernet0/0/0/19
!
!
cef load-balancing --->For load-balancing
fields l3 global
!

router ospf MPLS_CORE
area 200
interface Loopback0
!
interface GigabitEthernet0/0/0/19
!
interface GigabitEthernet0/0/1/9
!
!
!

//BNG - PPPoE Configuration

pppoe bba-group PPPoE-BBA-GRP1
service selection disable
!
class-map type control subscriber match-all PPPOE_CLASS
match protocol ppp
end-class-map
!
!
policy-map type control subscriber PPPOE_POLICY
event session-start match-first
class type control subscriber PPPOE_CLASS do-all
1 activate dynamic-template PPPOE_TEMPLATE
!
!
event session-activate match-first
class type control subscriber PPPOE_CLASS do-until-failure
1 authenticate aaa list default
!
!
end-policy-map
!
end
dynamic-template
type ppp PPPOE_TEMPLATE
ppp chap hostname ASR9k_BNG_PE1
ppp authentication chap pap
keepalive 60
!
!
interface Bundle-Ether50
bundle maximum-active links 1
!
interface Bundle-Ether50.1
vrf CPE_1_VRF_1
service-policy type control subscriber PPPOE_POLICY
pppoe enable bba-group PPPoE-BBA-GRP1
encapsulation dot1q 2
!
```

- PE2 Configuration

```
//VRF Configuration
```

```
vrf CPE_1_VRF_1
 address-family ipv4 unicast
  import route-target
   200:1
   200:3
   200:4
  !
  export route-target
   200:1
  !
 !
!

//BGP Configuration

router bgp 200
 address-family ipv4 unicast
  redistribute connected
 !
 address-family vpnv4 unicast
 !
 neighbor 65.0.0.1
  remote-as 200
  update-source Loopback0
  address-family vpnv4 unicast
  !
 !
 neighbor 65.0.0.3
  remote-as 200
  update-source Loopback0
  address-family vpnv4 unicast
  !
 !

//label-mode configuration

vrf CPE_1_VRF_1
  rd 65002:1
  address-family ipv4 unicast
   label mode per-prefix
   redistribute connected
  !
  neighbor 101.0.0.1
   remote-as 65535
   ebgp-multihop 5
   update-source Loopback1
   address-family ipv4 unicast
    route-policy EBGP_ROUTE_POLICY in
    route-policy EBGP_ROUTE_POLICY out
   !
  neighbor 102.0.0.1
   remote-as 65535
   ebgp-multihop 5
   update-source Loopback1
   address-family ipv4 unicast
    route-policy EBGP_ROUTE_POLICY in
    route-policy EBGP_ROUTE_POLICY out
   !
  !
 !

//MPLS Configuration
mpls ldp
 log
```

```
  neighbor
 !
 router-id 65.0.0.2 --->local
 interface GigabitEthernet0/2/1/1 --->connected to PE3
 !
  interface GigabitEthernet0/2/1/19 --->connected to PE1

 !
!
cef load-balancing
 fields l3 global
!

router ospf CORE
 area 200
  interface Loopback0
  !
  interface GigabitEthernet0/2/1/1
  !
  interface GigabitEthernet0/2/1/19
  !
 !
!

//BNG - PPPoE Configuration

interface Bundle-Ether60
!
interface Bundle-Ether60.1
 vrf CPE_1_VRF_1
 service-policy type control subscriber PPPOE_POLICY
 pppoe enable bba-group PPPoE-BBA-GRP1
 encapsulation dot1q 2
!
pppoe bba-group PPPoE-BBA-GRP1
 service selection disable
!
class-map type control subscriber match-all PPPOE_CLASS
 match protocol ppp
 end-class-map
!
!

policy-map type control subscriber PPPOE_POLICY
 event session-start match-first
  class type control subscriber PPPOE_CLASS do-all
   1 activate dynamic-template PPPOE_TEMPLATE
  !
 !
 event session-activate match-first
  class type control subscriber PPPOE_CLASS do-until-failure
   1 authenticate aaa list default
  !
 !
 end-policy-map
!
```

- PE3 Configuration

```
//VRF Configuration

vrf CPE_3_VRF_1
```

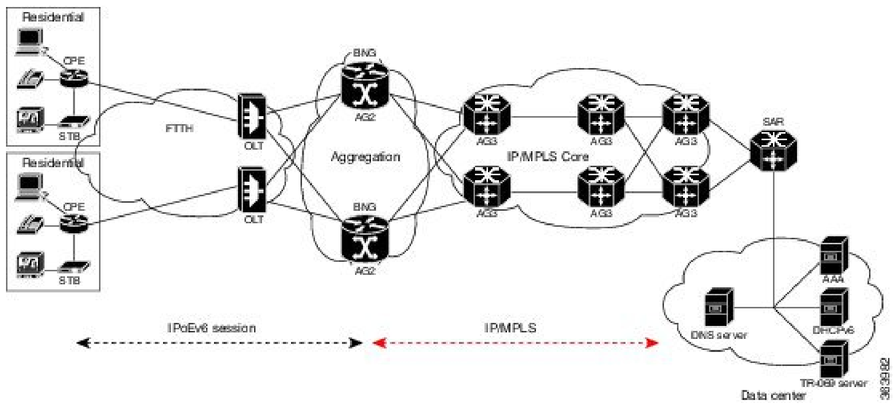
```
 address-family ipv4 unicast
  import route-target
   200:1
   200:2
   200:4
  !
  export route-target
   200:4
  !
 !
!

//BGP Configuration

router bgp 200
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 65.0.0.1
  remote-as 200
  update-source Loopback0
  address-family vpnv4 unicast
  !
 !
 neighbor 65.0.0.2
  remote-as 200
  update-source Loopback0
  address-family vpnv4 unicast
  !
 !

//maximum-paths and label-mode configuration

 vrf CPE_3_VRF_1
  rd 65003:1
  address-family ipv4 unicast
   label mode per-prefix
   maximum-paths ibgp 8 unequal-cost
  !
  neighbor 103.0.0.1
   remote-as 102
   ebgp-multihop 5
   update-source Loopback1
   address-family ipv4 unicast
    route-policy PASS_ALL_POLICY in
    route-policy PASS_ALL_POLICY out
   !
  !
 !
!

//MPLS Configuration

mpls ldp
 router-id 65.0.0.3
 interface GigabitEthernet0/0/0/9
 !
 interface GigabitEthernet0/1/0/9
 !
!
```

Routed Subscriber Sessions

Routed Subscriber Deployment Topology and Use Cases

This figure depicts a sample deployment topology for routed subscriber sessions:

Figure 41: Sample Deployment Topology for Routed Subscriber Sessions



This table lists some of the use cases supported for routed subscriber sessions in BNG:

Description	**Stack**
Off-Box DHCP, with access and subscriber in the same VRF on BNG, and static cover route for the subscriber subnet.	IPv4 or IPv6
Off-Box DHCP, with access and subscriber in cross or different VRF on BNG, and static cover route for the subscriber subnet.	IPv4 or IPv6
Standalone DHCPv6 proxy on BNG, with access and subscriber in the default VRF, and cover route added by DHCPv6 for PD prefixes pointing to LL address of CPE.	IPv6
Standalone DHCPv6 proxy on BNG, with access in the default VRF, and subscriber in the non-default VRF; cover route added by DHCPv6 for PD prefixes pointing to LL address of CPE.	IPv6
Standalone DHCPv6 server on BNG, with access and subscriber in the default VRF; cover route added by DHCPv6 for PD prefixes pointing to LL address of CPE.	IPv6

Description	Stack
Standalone DHCPv6 server on BNG, with access in the default VRF, and subscriber in the non-default vrf; cover route added by DHCPv6 for PD prefixes pointing to LL address of CPE.	IPv6

These use cases are not supported for routed subscriber sessions in BNG:

Description	Stack
On-Box DHCP standalone proxy or server, with access in the default VRF, and subscriber in the non-default VRF on BNG; /32 cover route added by DHCP, and the first hop router is proxy or relay.	IPv4
On-Box DHCP standalone proxy or server, with access and subscriber in the same VRF (default or non-default) on BNG.	IPv4
Off-Box DHCP, with access and subscriber in the same VRF (default or non-default);subscriber prefix-length matching with cover-route prefix-length is not supported.	IPv4 or IPv6

Sample Configurations for Routed Subscriber Session

This section provides the sample configurations for a use case scenario of packet-triggered routed subscriber session in BNG.

These are the sample configurations:

```
//Interface Configuration:

interface Bundle-Ether1
 [bundle load-balancing hash src-ip] --->optional
 lacp switchover suppress-flaps 2500
 bundle wait-while 1
 dampening 4
 bundle maximum-active links 2
!
interface Bundle-Ether1.201
 ipv4 address 15.15.15.1 255.255.255.0
 ipv6 address 15:15:15::1/64
 service-policy type control subscriber PL
 encapsulation dot1q 201
 ipsubscriber ipv4 routed
  initiator unclassified-ip
 !
 ipsubscriber ipv6 routed
  initiator unclassified-ip
 !
!

//Class-map Configuration:
```

```
class-map type control subscriber match-any ISN_CM_V6_1
 match source-address ipv6 2004:1:1::/48
 end-class-map
!
class-map type control subscriber match-any ISN_CM_V4_1
 match source-address ipv4 14.0.0.1 255.0.0.0
 end-class-map
!

//Dynamic Template Configuration:
dynamic-template
 type ipsubscriber ISN_TEMPLATE_V6_4
  ipv6 enable
 !
 type ipsubscriber ISN_TEMPLATE_V4_1
  ipv4 unnumbered Loopback1
 !
 type service httpr_service_temp_coa
  service-policy type pbr httpr-redirect-policy
 !

//Policy-map Configuration:

policy-map type control subscriber p_map_cntl_1
 event session-start match-all
  class type control subscriber ISN_CM_V6_1 do-until-failure
   1 activate dynamic-template ISN_TEMPLATE_V6_1
   2 authorize aaa list default format VID password cisco123
  !
  class type control subscriber ISN_CM_V4_1 do-until-failure
   1 activate dynamic-template ISN_TEMPLATE_V4_1
   2 authorize aaa list default format VID password cisco123
  !
 !
 event authorization-failure match-all
  class type control subscriber ISN_CM_V6_1 do-until-failure
   1 activate dynamic-template httpr_service_temp_coa
   2 set-timer T1 60
 class type control subscriber ISN_CM_V4_1 do-until-failure
   1 activate dynamic-template httpr_service_temp_coa
   2 set-timer T1 60
  !
 !

event account-logon match-all
  class type control subscriber ISN_CM_V6_1 do-until-failure
   1 authenticate aaa list default
   2 stop-timer T1
   3 deactivate dynamic-template httpr_service_temp_coa
  !
  class type control subscriber ISN_CM_V4_1 do-until-failure
   1 authenticate aaa list default
   2 stop-timer T1
   3 deactivate dynamic-template httpr_service_temp_coa
  !
 event account-logoff match-all
  class type control subscriber ISN_CM_V6_1 do-until-failure
   1 disconnect
  !
  class type control subscriber ISN_CM_V4_1 do-until-failure
   1 disconnect
  !
 !
```

```
event timer-expiry match-all
  class type control subscriber ISN_CM_V6_1 do-until-failure
   11 disconnect
  !
  class type control subscriber ISN_CM_V4_1 do-until-failure
   11 disconnect
   !
 end-policy-map
!

lpts punt police location 0/0/CPU0
 protocol unclassified rate 75
!
lpts punt police location 0/1/CPU0
 protocol unclassified rate 75
!

//Static Route Configuration:

router static
 address-family ipv4 unicast
  8.0.0.0/8 8.44.0.1
13.0.0.0/8 13.0.0.2
14.0.0.0/16 12.0.0.2 ---> summary route to subscriber network

dhcp ipv6
 profile pf1 server
  lease 0 0 10
  prefix-pool p1
 !
 profile pf3 proxy
  helper-address vrf red 2003::2
 !
 interface Bundle-Ether1.1 server profile pf1
 interface Bundle-Ether2.1 proxy profile pf3
!
pool vrf default ipv6 p1
 prefix-length 56
 prefix-range 2004:1:1:100:: 2004:1:1:100::
!

//RADIUS Configuration:

radius-server host 8.45.12.251 auth-port 1812 acct-port 1813
 key 7 094F471A1A0A
!
aaa server radius dynamic-author
 port 1700
 client 8.45.12.251 vrf default
  server-key 7 02050B5A
 !
!
radius-server source-port extended
aaa accounting network default start-stop group radius
aaa accounting service default group radius
aaa accounting subscriber default group radius
aaa authorization subscriber default group radius
!
```

Verification of Routed Subscriber Session Configurations

These show commands can be used to verify the routed subscriber session configurations in BNG.

SUMMARY STEPS

1. **show ipsubscriber access-interface**
2. **show ipsubscriber summary**
3. **show ipsubscriber interface brief**
4. **show ipsubscriber interface**
5. **show ipsubscriber interface**
6. **show subscriber session all summary**
7. **show subscriber session filter**
8. **show subscriber session filter**

DETAILED STEPS

Step 1 **show ipsubscriber access-interface**

Displays the access-interface information for IP subscriber.

Example:

```
RP/0/RSP0/CPU0:router#
show ipsubscriber access-interface bundle-Ether 1.201

---
---
Mon Sep  1 18:05:15.899 UTC
Interface: Bundle-Ether1.201
  State: UP
  Type: Plain
  Interface Type: Routed
  Created Sep  1 17:54:17 (age 00:10:58)
  Initiator DHCP disabled
    Session count 0
    FSOL packets 0
    FSOL dropped packets 0
    FSOL flow rate dropped packets 0
    FSOL session limit dropped packets 0
  Initiator Packet-Trigger enabled
    Session count 1
    FSOL packets 3, bytes 300
    FSOL dropped packets 2, bytes 200
    FSOL flow rate dropped packets 0
    FSOL session limit dropped packets 0
  Initiator DHCPv6 disabled
    Session count 0
    FSOL packets 0
    FSOL dropped packets 0
    FSOL flow rate dropped packets 0
    FSOL session limit dropped packets 0
  Initiator Packet-Trigger-IPv6 enabled
    Session count 1
    FSOL packets 1, bytes 100
    FSOL dropped packets 0, bytes 0
    FSOL flow rate dropped packets 0
    FSOL session limit dropped packets 0
  Session limits per-vlan
    All sources 0
    Unclassified-source 0
```

Step 2 **show ipsubscriber summary**

Displays the summary information for IP subscriber interfaces.

Example:

```
RP/0/RSP0/CPU0:router#
show ipsubscriber summary

Mon Sep  1 18:05:48.610 UTC
IPSUB Summary for all nodes

Interface Counts:
                                    DHCP  Pkt Trigger
                              ---------- ------------
                    Invalid:          0            0
                Initialized:          0            0
    Session creation started:         0            0
    Control-policy executing:         0            0
     Control-policy executed:         0            0
    Session features applied:         0            0
              VRF configured:         0            0
            Adding adjacency:         0            0
             Adjacency added:         0            0
                         Up:          0            1
                       Down:          0            0
                    Down AF:          0            0
           Down AF Complete:          0            0
              Disconnecting:          0            0
               Disconnected:          0            0
                      Error:          0            0
                              ---------- ------------
                      Total:          0            1

                                  DHCPv6 PktTrig-IPv6
                              ---------- ------------
                    Invalid:          0            0
                Initialized:          0            0
    Session creation started:         0            0
    Control-policy executing:         0            0
     Control-policy executed:         0            0
    Session features applied:         0            0
              VRF configured:         0            0
            Adding adjacency:         0            0
             Adjacency added:         0            0
                         Up:          0            1
                       Down:          0            0
                    Down AF:          0            0
           Down AF Complete:          0            0
              Disconnecting:          0            0
               Disconnected:          0            0
                      Error:          0            0
                              ---------- ------------
                      Total:          0            1

Routes Per VRF (1 VRFs) [Packet-Trigger]:
                              IPv4 Count IPv6 Count
                              ---------- ----------
                    default:          1          1

Access Interface Counts (1 interfaces):

                                    DHCP  Pkt Trigger
                              ---------- ------------
```

BNG Use Cases and Sample Configurations

```
                FSOL Packets:          0           3
                  FSOL Bytes:          0         300

                                  DHCPv6 PktTrig-IPv6
                              ---------- ------------
                FSOL Packets:          0           1
                  FSOL Bytes:          0         100
```

Step 3 **show ipsubscriber interface brief**

Displays the brief summary of IP Subscriber access-interface status and configuration.

Example:

```
RP/0/RSP0/CPU0:router#
show ipsubscriber interface brief

Mon Sep  1 18:06:33.713 UTC
Codes: INV - Invalid, INIT - Initialized, STRTD - Session Creation Started,
       CPEXCTG - Control-Policy Executing, CPEXCTD - Control-Policy Executed,
       FTAPPLD - Session Features Applied, VRFCFGD - VRF Configured,
       ADJADDG - Adding Adjacency, ADJADDD - Adjacency Added, UP - Up,
       DOWN - Down, DISCG - Disconnecting, DISCD - Disconnected, ERR - Error,
       UNKWN - Unknown State, PKT - Packet Trigger Initiation,
       PKTv6 - Packet Trigger Initiation for IPv6,
       DHCP - DHCP Initiation, DHCPv6 - DHCPv6 Initiation

Interface              Proto  Subscriber IP      MAC Address        Sublabel   VRF        State

---------------------- ------ ------------------ ------------------ ---------- ---------- ----------
BE1.201.ip1            PKT    1.10.10.1          0000.0000.0003     0x41       default    UP

BE1.201.ip2            PKTv6                     0001.0001.0000     0xc3       default    UP
```

Step 4 **show ipsubscriber interface**

Displays the interface information for the IP subscriber interfaces.

Example:

```
RP/0/RSP0/CPU0:router#
show ipsubscriber interface Bundle-Ether 1.201.ip1

Mon Sep  1 18:06:54.213 UTC
Interface: Bundle-Ether1.201.ip1
  Type: Routed
  Access Interface: Bundle-Ether1.201
  Subscriber IPv4: 1.10.10.1
  Subscriber Label: 0x41
  IPv4 Initiator: Packet-Trigger
  VLAN ID: 201
  Created: Sep  1 17:58:24 (age 00:08:30)
  VRF: default, IPv4 Table: default
  IPv4 State: Up (old: Adjacency added)
       Last state change: Sep  1 17:58:25 (00:08:29 in current state)
```

Step 5 **show ipsubscriber interface**

Displays the interface information for the IP subscriber interfaces.

Example:

```
RP/0/RSP0/CPU0:router#
show ipsubscriber interface Bundle-Ether 1.201.ip2

Mon Sep  1 18:06:57.846 UTC
Interface: Bundle-Ether1.201.ip2
  Type: Routed
  Access Interface: Bundle-Ether1.201
  Subscriber IPv6: 2001:0:1:1::1
  Subscriber IPv6 Prefix: 2001:0:1:1::/64
  Subscriber Label: 0xc3
  IPv6 Initiator: Packet-Trigger-IPv6
  VLAN ID: 201
  Created: Sep  1 17:58:59 (age 00:07:58)
  VRF: default, IPv6 Table: default
  IPv6 State: Up (old: Adjacency added)
       Last state change: Sep  1 17:59:00 (00:07:57 in current state)
```

Step 6 **show subscriber session all summary**

Displays the session summary information for all nodes.

Example:

```
RP/0/RSP0/CPU0:router#
show subscriber session all summary

Mon Sep  1 18:07:29.791 UTC

Session Summary Information for all nodes

                Type            PPPoE           IPSub           IPSub
                                                (DHCP)          (PKT)
                ====            =====           ======          =====

Session Counts by State:
        initializing            0               0               0
          connecting            0               0               0
           connected            0               0               0
           activated            0               0               2
                idle            0               0               0
       disconnecting            0               0               0
                 end            0               0               0
              Total:            0               0               2


Session Counts by Address-Family/LAC:
         in progress            0               0               0
           ipv4-only            0               0               1
           ipv6-only            0               0               1
     dual-partial-up            0               0               0
             dual-up            0               0               0
                 lac            0               0               0
              Total:            0               0               2
```

Step 7 **show subscriber session filter**

Displays the subscriber management session information based on the filter criteria.

Example:

```
RP/0/RSP0/CPU0:router#
show subscriber session filter interface bundle-ether 1.201.ip1 detail


Mon Sep  1 18:09:51.381 UTC
Interface:                Bundle-Ether1.201.ip1
Circuit ID:               Unknown
Remote ID:                Unknown
Type:                     IP: Packet-trigger
IPv4 State:               Up, Mon Sep  1 17:58:25 2014
IPv4 Address:             1.10.10.1, VRF: default
Mac Address:              Unknown
Account-Session Id:       0001137f
Nas-Port:                 Unknown
User name:                unknown
Outer VLAN ID:            201
Subscriber Label:         0x00000041
Created:                  Mon Sep  1 17:58:24 2014
State:                    Activated
Authentication:           unauthenticated
Authorization:            unauthorized
Access-interface:         Bundle-Ether1.201
Policy Executed:
policy-map type control subscriber PL
  event Session-Start match-first [at Mon Sep  1 17:58:24 2014]
    class type control subscriber class-default do-all [Succeeded]
      10 activate dynamic-template ptrigger [Succeeded]
Session Accounting: disabled
Last COA request received: unavailable
```

Step 8 **show subscriber session filter**

Displays the subscriber management session information based on the filter criteria.

Example:

```
RP/0/RSP0/CPU0:router#
show subscriber session filter interface bundle-ether 1.201.ip2 detail


Mon Sep  1 18:10:45.883 UTC
Interface:                Bundle-Ether1.201.ip2
Circuit ID:               Unknown
Remote ID:                Unknown
Type:                     IP: Packet-trigger
IPv6 State:               Up, Mon Sep  1 17:59:00 2014
IPv6 Address:             2001:0:1:1::1, VRF: default
IPv6 Interface ID:        ........ (00 00 00 00 00 00 00 01)
Mac Address:              Unknown
Account-Session Id:       00011380
Nas-Port:                 Unknown
User name:                unknown
Outer VLAN ID:            201
Subscriber Label:         0x000000c3
Created:                  Mon Sep  1 17:58:59 2014
State:                    Activated
Authentication:           unauthenticated
Authorization:            unauthorized
Access-interface:         Bundle-Ether1.201
Policy Executed:
policy-map type control subscriber PL
  event Session-Start match-first [at Mon Sep  1 17:58:59 2014]
```

BNG Use Cases and Sample Configurations

```
    class type control subscriber class-default do-all [Succeeded]
      10 activate dynamic-template ptrigger [Succeeded]
Session Accounting: disabled
Last COA request received: unavailable
```

APPENDIX **E**

DIAMETER Attributes

BNG Supports DIAMETER Gx interface for Policy and Charging Provisioning with the PCRF, and DIAMETER Gy interface for Online Charging Service with OCS.

This Appendix lists the applicable AVPs in each Diameter Request that is sent or received by BNG, and also some sample packets.

BNG DIAMETER Gx Application AVPs

The DIAMETER interface with BNG is based on the respective latest 3GPP specifications and Diameter Credit Control Application (RFC 4006) standard. The interface remains the same for any DIAMETER server vendor, but the user must be aware of the set of AVPs being used to address the BNG-DIAMETER deployments and use-cases. This topic lists the BNG DIAMETER Gx Application AVPs.

Table 26: BNG DIAMETER Gx Application AVPs

AVP Name	AVP Code	Vendor	AVP Format	Messages	Source	Flags
Session-ID	263	IETF	String	CCR, CCA	RFC 6733	M
Origin-Host-Name	264	IETF	String	CCR, CCA	RFC 6733	M
Origin-Realm	296	IETF	String	CCR, CCA	RFC 6733	M
Destination Realm	283	IETF	String	CCR, CCA	RFC 6733	M
CC-Request-Type	416	IETF	Ulong	CCR, CCA	RFC 4006	M
CC-Request-Number	415	IETF	Ulong	CCR, CCA	RFC 4006	M
Auth-Application-ID	258	IETF	Ulong	CCR, CCA	RFC 6733	M

AVP Name	AVP Code	Vendor	AVP Format	Messages	Source	Flags
ReAuth-Request-Type	285	IETF	Ulong	RAR	RFC 6733	M
Username	1	IETF	String	CCR, CCA	RFC 6733	M
Framed-IP-Address	8	IETF	Address	CCR, CCA	RFC 6733	M
Logical-Access-Id	302	ETSI	OctectString	CCR-I	TS 29.212	
Physical-Access-Id	313	ETSI	OctectString	CCR-I	TS 29.212	
*User-Equipment-Info	458	IETF	Grouped	CCR-I	RFC 6733	
User-Equipment-Info-Type	459	IETF	Enum	CCR-I	RFC 6733	M
User-Equipment-Info-Value	460	IETF	String	CCR-I	RFC 6733	M
Result-Code	268	IETF	Enum	CCA	RFC 6733	M
*Charging-Rule-Install	1001	3GPP	Grouped	CCA, RAR	TS 29.212	M, V
*Charging-Rule-Definition	1003	3GPP	Grouped	CCA, RAR	TS 29.212	M, V
Charging-Rule-Name	1004	3GPP	OctectString	CCA, RAR	TS 29.212	M, V
CC-Service-Identifier	439	IETF	Ulong	CCA, RAR	RFC 4006	M
Rating-Group	432	IETF	Ulong	CCA, RAR	RFC 4006	M
*Charging-Rule-Remove	1002	3GPP	Grouped	RAR	TS 29.212	
Session-Release-Cause	1045	3GPP	Enum	RAR	TS 29.212	M
Gx-Application-ID	14080	Cisco	Ulong	CCR		
Vrf-Id	14002	Cisco	Ulong	CCA-I		
IPv6-Enable	14056	Cisco	Ulong	CCA-I		
IP-Unnumbered	14055	Cisco	String	CCA-I		
Inacl	14009	Cisco	String	CCA-I		
Ipv6-inacl	14010	Cisco	String	CCA-I		
Outacl	14012	Cisco	String	CCA-I		
IPv6-outacl	14013	Cisco	String	CCA-I		
Sub-Qos-Policy-In	14014	Cisco	String	CCA-I		
Sub-Qos-Policy-Out	14015	Cisco	String	CCA-I		
Accounting-List	14016	Cisco	String	CCA-I		

BNG DIAMETER Gy Application AVPs

This table lists the BNG DIAMETER Gy application AVPs.

Table 27: BNG DIAMETER Gy Application AVPs

AVP Name	AVP Code	Vendor	AVP Format	Messages	Source	Flags
Session-ID	263	IETF	String	CCR, CCA	RFC 6733	M
Origin-Host-Name	264	IETF	String	CCR, CCA	RFC 6733	M
Origin-Realm	296	IETF	String	CCR, CCA	RFC 6733	M
Destination Realm	283	IETF	String	CCR, CCA	RFC 6733	M
CC-Request-Type	416	IETF	Ulong	CCR, CCA	RFC 4006	M
CC-Request-Number	415	IETF	Ulong	CCR, CCA	RFC 4006	M
Auth-Application-ID	258	IETF	Ulong	CCR, CCA	RFC 6733	M
ReAuth-Request-Type	285	IETF	Ulong	RAR	RFC 6733	M
Username	1	IETF	String	CCR, CCA	RFC 6733	M
Framed-IP-Address	8	IETF	Address	CCR, CCA	RFC 6733	M
*User-Equipment-Info	458	IETF	Grouped	CCR-I	RFC 6733	-
User-Equipment-Info-Type	459	IETF	Enum	CCR-I	RFC 6733	M
User-Equipment-Info-Value	460	IETF	String	CCR-I	RFC 6733	M
Result-Code	268	IETF	Enum	CCA	RFC 6733	M
Service-Context-Id	461	IETF	String	CCR	RFC 4006	M
Event-Time-Stamp	55	IETF	Ulong	CCR	RFC 6733	M
CC-Multiple-Service-Support	455	IETF	Enum	CCR	RFC 4006	M
*CC-Multiple-Service	456	IETF	Grouped	CCR, CCA	RFC 4006	M
CC-Service-Identifier	439	IETF	Ulong	CCA, RAR	RFC 4006	M
Rating-Group	432	IETF	Ulong	CCA, RAR	RFC 4006	M
*Requested-Service-Unit	437	IETF	Ulong	CCR-I	RFC 4006	-
CC-Session-Failover	418	IETF	Ulong	CCA	RFC 4006	-
*Granted-Service-Unit	431	IETF	Grouped	CCA	RFC 4006	M
CC-Tariff-Time-Change	451	IETF	Ulong	CCR-U, CCA	RFC 4006	M

AVP Name	AVP Code	Vendor	AVP Format	Messages	Source	Flags
CC-Time	420	IETF	Ulong	CCR, CCA	RFC 4006	M
CC-Input-Octets	412	IETF	Ulonglong	CCR, CCA	RFC 4006	M
CC-Output-Octets	414	IETF	Ulonglong	CCR, CCA	RFC 4006	M
CC-Total-Octets	421	IETF	Ulonglong	CCR, CCA	RFC 4006	M
CC-Tariff-Change-Units	446	IETF	Enum	CCR-U	RFC 4006	M
Reporting-Reason	872	IETF	Enum	CCR-U	RFC 4006	M
Volume-Quota-Threshold	869	3GPP	Ulong	CCA	TS 32.299	M, V
Time-Quota-Threshold	868	3GPP	Ulong	CCA	TS 32.2996	M, V
CC-Validity-Time	448	IETF	Ulong	CCA	RFC 4006	M
Quota-Holding-Time	871	3GPP	Ulong	CCA	RFC 4006	M
Credit-Control-Failure-Handling	427	IETF	Enum	CCA	RFC 4006	M
*Final-Unit-Indication	430	IETF	Grouped	CCA	RFC 4006	M
Final-Unit-Action	449	IETF	Enum	CCA	RFC 4006	M
Termination-Cause	295	IETF	Enum	CCR-Final	RFC 6733	M

BNG DIAMETER NASREQ Application Cisco AVPs

This table lists the BNG DIAMETER NASREQ application Cisco AVPs.

Table 28: BNG DIAMETER NASREQ Application Cisco AVPs

AVP Name	Value	Format	Type
Framed-IP-Address	8	ipv4addr	ietf
Framed-IP-Address/ Framed-IP-Netmask	9	ipv4addr	ietf
Filter-Id	11	binary	ietf
Framed-MTU	12	ulong	ietf
Framed-Compression	13	enum	ietf
Reply-Message	18	binary	ietf
Framed-Route	22	string	ietf
Session-Timeout	27	ulong	ietf

AVP Name	Value	Format	Type
Idle-Timeout	28	ulong	ietf
Framed-Pool	88	string	ietf
Framed-IPv6-Prefix	97	binary	ietf
Framed-IPv6-Route	99	string	ietf
Framed-IPv6-Pool	100	string	ietf
Delegated-IPv6-Prefix	123	binary	ietf
ip:primary-dns / ip:secondary-dns	135	address	ietf
addrv6	14001	address	cisco_vsa
vrf-id	14002	ulong	cisco_vsa
parent-session-id	14003	string	cisco_vsa
service-name	14004	string	cisco_vsa
disc-cause-ext	14005	enum	cisco_vsa
disconnect-cause	14006	string	cisco_vsa
Ascend-Connect-Progress	14007	ulong	cisco_vsa
Acct-Unique-Session-Id	14008	ulong	cisco_vsa
inacl	14009	string	cisco_vsa
ipv6_inacl	14010	string	cisco_vsa
cisco-nas-port	14011	string	cisco_vsa
outacl	14012	string	cisco_vsa
ipv6_outacl	14013	string	cisco_vsa
sub-qos-policy-in	14014	string	cisco_vsa
sub-qos-policy-out	14015	string	cisco_vsa
accounting-list	14016	string	cisco_vsa
parent-if-handle	14017	ulong	cisco_vsa
acct-input-gigawords-ipv4	14018	ulong	cisco_vsa
acct-input-octets-ipv4	14019	ulong	cisco_vsa
acct-input-packets-ipv4	14020	ulong	cisco_vsa
acct-output-gigawords-ipv4	14021	ulong	cisco_vsa

DIAMETER Attributes

AVP Name	Value	Format	Type
acct-output-octets-ipv4	14022	ulong	cisco_vsa
acct-output-packets-ipv4	14023	ulong	cisco_vsa
acct-input-gigawords-ipv6	14024	ulong	cisco_vsa
acct-input-octets-ipv6	14025	ulong	cisco_vsa
acct-input-packets-ipv6	14026	ulong	cisco_vsa
acct-output-gigawords-ipv6	14027	ulong	cisco_vsa
acct-output-octets-ipv6	14028	ulong	cisco_vsa
acct-output-packets-ipv6	14029	ulong	cisco_vsa
subscriber:command=account-logon	14030	string	cisco_vsa
subscriber:sd=service1	14031	string	cisco_vsa
subscriber:sa=service1	14032	string	cisco_vsa
subscriber:sm= svc1(interim-interval=120)	14033	string	cisco_vsa
ip:ip-unnumbered=<loopback>	14038	string	cisco_vsa
ipv4:ipv4-multicast=Qos Correlation	14039	enum	cisco_vsa
ip:keepalive	14040	string	cisco_vsa
dual-stack-delay	14041	string	cisco_vsa
idle-timeout-direction	14042	string	cisco_vsa
idlethreshold	14043	ulong	cisco_vsa
ipv4:ipv4-mtu	14044	ulong	cisco_vsa
ipv6:ipv6-mtu	14045	ulong	cisco_vsa
md-ip-addr	14046	address	cisco_vsa
md-port	14047	ulong	cisco_vsa
md-dscp	14048	ulong	cisco_vsa
li-action	14049	ulong	cisco_vsa
intercept-id	14050	binary	cisco_vsa
cisco-mpc-protocol-interface=pmipv6	14051	enum	cisco_vsa
cisco-mobile-node-identifier	14052	string	cisco_vsa
cisco-mn-service	14053	enum	cisco_vsa

AVP Name	Value	Format	Type
home-lma	14054	string	cisco_vsa

DIAMETER Accounting AVP

This table lists the DIAMETER Accounting AVPs that describe accounting usage information related to a specific session and for a service.

AVP	Command Code	Description
Accounting-Record-Type	480	Contains the type of accounting record being sent. This is similar to Acct-Status-Type RADIUS IETF AVP. These are the values currently defined for the Accounting-Record-Type AVP: • 1 - EVENT_RECORD • 2 - START_RECORD • 3 - INTERIM_RECORD • 4 - STOP_RECORD
Accounting-Record-Number	485	Identifies the record within one session. For a given BNG session, START_RECORD carries the value 0 (zero) and it increases for the subsequent Accounting-Request for the same BNG Session.
Accounting-Sub-Session-Id	287	Contains the accounting sub-session identifier. The combination of the Session-Id and this AVP must be unique for each sub-session, and the value of this AVP must be increased by one for all new sub-sessions. **Note** This is not supported in BNG.

AVP Considerations

These AVPs are considered to be security-sensitive:

- Acct-Interim-Interval

- Accounting-Realtime-Required
- Acct-Multi-Session-Id
- Accounting-Record-Number
- Accounting-Record-Type
- Accounting-Session-Id
- Accounting-Sub-Session-Id
- Class
- Session-Id
- Session-Binding
- Session-Server-Failover
- User-Name

DIAMETER Session-Id AVP

The Session-Id AVP (AVP code 263) is of type UTF8String and is used to identify a specific diameter application session. All messages pertaining to a given BNG subscriber session have the same Session-Id and uses the same value throughout the life of the session. The Session-Id AVP must appear immediately after the DIAMETER header and it optimizes the session association while parsing the request or response. The Session-Id AVP includes a mandatory portion and an implementation-specific portion.

The syntax for this AVP is:

DiameterIdentity;high 32 bits;low 32 bits;optional value

Table 29: Syntax Description

DiameterIdentity	Used to identify either: • A DIAMETER node for the purpose of duplicate connection and routing loop detection. • A realm to determine whether the messages can be processed locally or whether they must be routed or redirected.
high 32 bits *low 32 bits*	Decimal representation of the high and low 32 bits of a increasing 64-bit value. BNG uses Session-Id (Acct-Session-ID) which is of 32 bits as of now and hence the high 32 bits is 0 (Zero) to start with. To make the DIAMETER Session-Id unique, the high 32 bits is increased every time after a router reload or in process restart scenarios (such as DIAMETER process restart) where the previous Session-Id is lost. This eliminates the possibility of overlapping Session-Ids in such scenarios.

optional value	This is implementation specific, and it may include a Layer 2 address, timestamp and so on. BNG uses the *timestamp* that is retrieved when the first request is being sent to the DIAMETER server for each application. The first message for NASREQ is AA-Request, and for DCCA application, it is CCR-Initial request. The subsequent messages carry the same timestamp for the given BNG subscriber session to ensure that the same Session-Id is used in the entire span of the respective DIAMETER application session.

This is an example of DIAMETER Session-Id AVP (with an optional value):

```
BNG1.example.com;1876543210;523;BNG1@10.0.0.1
```

This is an example of DIAMETER Session-Id AVP (without an optional value):

```
BNG1.example.com;1876543210;523
```

RADIUS Attributes in DIAMETER Messages

A DIAMETER message may include RADIUS attributes, except the ones listed in this table. These RADIUS attributes, if present, are translated internally to similar DIAMETER AVPs.

RADIUS Attribute	RADIUS Attribute Name	Nearest DIAMETER AVP
3	CHAP-Password	CHAP-Auth Group
26	Vendor-Specific	Vendor Specific AVP
29	Termination-Action	Authorization-Lifetime
40	Acct-Status-Type	Accounting-Record-Type
42	Acct-Input-Octets	Accounting-Input-Octets
43	Acct-Output-Octets	Accounting-Output-Octets
47	Acct-Input-Packets	Accounting-Input-Packets
48	Acct-Output-Packets	Accounting-Output-Packets
49	Acct-Terminate-Cause	Termination-Cause
52	Acct-Input-Gigawords	Accounting-Input-Octets
53	Acct-Output-Gigawords	Accounting-Output-Octets
80	Message-Authenticator	No corresponding DIAMETER AVP. If this attribute is present, it is checked and discarded.

Sample Packets for BNG DIAMETER Messages

This topic lists the sample packets for BNG DIAMETER messages.

GX - CCR-Initial Message from BNG to PCRF

```
Session-Id                          [263]     "cisco123.com;3201010A;0"  (M)
  Origin-host-name                  [264]     "cisco123.com"             (M)
  Origin-Realm                      [296]     "cisco.com"                (M)
  CC-request-type                   [416]     ccr-initial                (M)
  CC-request-number                 [415]     0                          (M)
  Destination-Realm                 [283]     "cisco.com"                (M)
  Auth-Application-ID               [258]     16777238                   (M)
  User-name                         [1]       "prepaid-user"             (M)
  Framed-IP-Address                 [8]       10.0.0.1                   (M)
  User-Equipment-Info               [458]
  User-Equipment-Info-Type          [459]     MAC (1)                    (M)
     User-Equipment-Info-Value      [460]     "0219.a220.e809"             (M)
```

GX - CCA-Initial Message from PCRF to BNG

```
Session-Id                           [263]    "cisco123.com;3201010A;0"  (M)
 Result-code                         [268]    2001                       (M)
 Auth-Application-ID                 [258]    16777238                   (M)
 Origin-host-name                    [264]    "pcrf.cisco.com"           (M)
 Origin-Realm                        [296]    "cisco.com"                (M)
 Destination-Realm                   [283]    "cisco.com"                (M)
 CC-request-type                     [416]    ccr-initial                (M)
 CC-request-number                   [415]    0                          (M)
 Vendor, 3GENPP                      [10415]
 Charging-Rule-Install               [1001]                              (M)
     Vendor, 3GENPP                  [10415]
     Charging-Rule-Definition        [1003]                              (M)
         Vendor, 3GENPP              [10415]
         Charging Rule Name          [1005]   "HSI_10MB"                 (M)
         CC-service-identifier       [439]     5                         (M)
         Rating group                [432]     3                         (M)
```

Gx - Application RA-Request Message

```
Session-Id                            [263]    "cisco123.com;3201010A;0"  (M)
  Origin-host-name                    [264]    "cisco123.com"             (M)
  Origin-Realm                        [296]    "cisco.com"                (M)
  Re-Auth-Request-Type                [285]    AUTHORIZE_ONLY (0)         (M)
  CC-request-number                   [415]    0                          (M)
  Destination-Realm                   [283]    "cisco.com"                (M)
  Auth-Application-ID                 [258]    Gx  (16777238)
  Vendor, 3GENPP                      [10415]                             (M)
  Charging Rule Remove                [1002]
       Vendor, 3GENPP                 [10415]
       Charging Rule Name             [1005]   "HSI_10MB"                 (M)
  Vendor, 3GENPP                      [10415]                             (M)
  Charging Rule Install               [1001]
      Vendor, 3GENPP                  [10415]
      Charging-Rule-Definition        [1003]                              (M)
          Vendor, 3GENPP              [10415]
```

```
         Charging Rule Name        [1005]   "LOW_BW128KB"              (M)
         CC-service-identifier     [439]    6                          (M)
         Vendor, 3GENPP            [10415]
         Rating group              [432]    4                          (M)
```

Gx - Application RA-Answer Message

```
Session-Id                         [263]    "cisco123.com;3201010A;0"   (M)
Result-code                        [268]    2001                        (M)
Auth-Application-ID                [258]    Gx (16777238)               (M)
Origin-host-name                   [264]    "pcrf.cisco.com"            (M)
Origin-Realm                       [296]    "cisco.com"                 (M)
Destination-Realm                  [283]    "cisco.com"                 (M)
```

Gx - RA-Request Message to Release the Session

```
Session-Id                         [263]     "cisco123.com;3201010A;0" (M)
Origin-host-name                   [264]     "cisco123.com"            (M)
Origin-Realm                       [296]     "cisco.com"               (M)
Re-Auth-Request-Type               [285]     AUTHORIZE_ONLY (0)        (M)
CC-request-number                  [415]     0                         (M)
Destination-Realm                  [283]     "cisco.com"               (M)
Auth-Application-ID                [258]     Gx  (16777238)
Vendor, 3GENPP                     [10415]
 Session-Release-Cause             [1045]    "UNSPECIFIED_REASON"      (M)
```

Gy - CCR-Initial Message

```
Session-Id                         [263]     "cisco123.com;3201010A;0"  (M)
Origin-host-name                   [264]     "cisco123.com"             (M)
Origin-Realm                       [296]     "cisco.com"                (M)
CC-request-type                    [416]     ccr-initial                (M)
CC-request-number                  [415]     0                          (M)
Destination-Realm                  [283]     "cisco.com"                (M)
Auth-Application-ID                [258]     4                          (M)
User-name                          [1]       "prepaid-user"             (M)
Service_Context_Id                 [461]     "32251@3gpp.org"              (M)
Framed-IP-Address                  [8]       10.0.0.1                   (M)
Event-Timestamp                    [55]      3426644002                 (M)
CC-Multiple-service-support        [455]     multiple-service-supported (M)
CC-multiple-service                [456]
    CC-service-identifier          [439]     5                          (M)
    Rating group                   [432]     3                          (M)
    CC-requested-service-unit      [437]
    User-Equipment-Info            [458]
  User-Equipment-Info-Type         [459]     MAC (1)                    (M)
       User-Equipment-Info-Value   [460]     "0219.a220.e809"             (M)
```

Gy - CCA-Initial Message

```
Session-Id                         [263]     "cisco123.com;3201010A;0"   (M)
Result-code                        [268]     2001                        (M)
Origin-host-name                   [264]     "cisco123.com"              (M)
Origin-Realm                       [296]     "cisco.com"                 (M)
CC-request-type                    [416]     ccr-initial                 (M)
CC-request-number                  [415]     0                           (M)
```

```
  Destination-Realm                  [283]     "cisco.com"                (M)
  Auth-Application-ID                [258]     4                          (M)
  User-name                          [1]       “prepaid-user”             (M)
  Framed-IP-Address                  [8]       10.0.0.1                   (M)
  Event-Timestamp                    [55]      3426644002                 (M)
  CC-session-failover                [418]     NOT_SUPPORTED (0)          (M)
  CC-multiple-service                [456]
   Granted-Service-Unit              [431]
    CC-Tariff-Time-Change            [451]     10:10:10  Thu Dec 28 2006  (M)
    CC-Total-Octets                  [414]     1000                       (M)
   Vendor, 3GENPP                    [10415]
   Volume-Quota-threshold            [869]     10
   Rating-Group                      [432]     3                          (M)
   Service-Identifier                [439]     5                          (M)
   CC-validity-time                  [448]     1000                       (M)
   Result-code                       [268]     2001                       (M)
   Vendor, 3GENPP                    [10415]
   Quota-Holding-Time                [871]     10000                      (M)
  Credit-Control-Failure-Handling    [427]     CONTINUE (1)               (M)
```

Gy - CCR-Update Message

```
Session-Id                      [263]    " cisco123.com;3201010A;0"     (M)
  Origin-host-name              [264]    "cisco123.com"                 (M)
  Origin-Realm                  [296]    "cisco.com"                    (M)
  Auth-Application-ID           [258]    4                              (M)
  CC-request-type               [416]    ccr-update                     (M)
  CC-request-number             [415]    1                              (M)
  Service_Context_Id            [461]    "bng1@cisco.com                (M)
  Framed-IP-Address             [8]      10.0.0.1                       (M)
  User-Name                     [1]      "prepaid-user"                 (M)
  Event-Timestamp               [55]     3426644119                     (M)
  Destination-Realm             [283]    "cisco.com"                    (M)
  CC-multiple-service           [456]
   CC-rating-group              [432]    3                              (M)
   Service-Identifier           [439]    5                              (M)
    Used-Service-Unit           [446]
    CC-Time                     [420]    1                              (M)
    Vendor, 3GENPP              [10415]
    Reporting- Reason           [872]    QUOTA_EXHAUSTED (3)            (M)
   Used-Service-Unit            [446]
    CC-Total-Octets             [421]    2                              (M)
    Vendor, 3GENPP              [10415]
    Reporting-Reason            [872]    QUOTA_EXHAUSTED (3)            (M)
    CC-Input-Octets             [412]    1                              (M)
    CC-Output-Octets            [414]    1                              (M)
   User-Equipment-Info          [458]
    User-Equipment-Info-Type    [459]    MAC (1)                        (M)
    User-Equipment-Info-Value   [460]    “0219.a220.e809”                 (M)
```

CCR-Update Message with Tariff Change Units

```
CC-multiple-service             [456]
  CC-rating-group               [432]    3                           (M)
  Service-Identifier            [439]    5                           (M)
  CC-Service-Unit               [446]
    CC-Time                     [420]    166                         (M)
    CC-tariff-change-units      [452]    units-before-tariff-change  (M)
  CC-Service-Unit               [446]
    CC-Total-Octets             [421]    264                         (M)
```

DIAMETER Attributes

```
    CC-Input-Octets             [412]    132                         (M)
    CC-Output-Octets            [414]    132                         (M)
    CC-tariff-change-units      [452]    units-before-tariff-change  (M)
  CC-Service-Unit               [446]
    CC-Time                     [420]    129                         (M)
    CC-tariff-change-units      [452]    units-after-tariff-change   (M)
  CC-Service-Unit               [446]
    CC-Total-Octets             [421]    132                         (M)
    CC-Input-Octets             [412]    66                          (M)
    CC-Output-Octets            [414]    66                          (M)
    CC-tariff-change-units      [452]    units-after-tariff-change   (M)
```

Gy - CCA-Update Message

```
Session-Id                      [263]     "cisco123.com;3201010A;0"  (M)
Result-code                     [268]     2001                       (M)
Origin-host-name                [264]     "cisco123.com"             (M)
Origin-Realm                    [296]     "cisco.com"                (M)
CC-request-type                 [416]     ccr-update                 (M)
CC-request-number               [415]     0                          (M)
Destination-Realm               [283]     "cisco.com"                (M)
Auth-Application-ID             [258]     4                          (M)
CC-session-failover             [418]     NOT_SUPPORTED (0)          (M)
CC-multiple-service             [456]
 Granted-Service-Unit           [431]
  CC-Tariff-Time-Change         [451]     10:10:10  Thu Dec 28 2006  (M)
  CC-Total-Octets               [414]     1000                       (M)
 CC-final-unit-indication       [430]
 Final-Unit-Action              [449]     0 (Terminate)              (M)
 Vendor, 3GENPP                 [10415]
 Volume-Quota-threshold         [869]     10
 Rating-Group                   [432]     3                          (M)
 Service-Identifier             [439]     5                          (M)
 Result-code                    [268]     2001                       (M)
 Vendor, 3GENPP                 [10415]
 Quota-Holding-Time             [871]     10000                      (M)
 Quota-Validity-Time            [448]     10000                      (M)
Credit-Control-Failure-Handling  [427]     CONTINUE (1)              (M)
```

Gy - Sample RAR Packet

```
Session-Id                    [263]     "bng;167;1234567"            (M)
Origin-host-name              [264]     "diameter1.cisco.com"        (M)
Origin-Realm                  [296]     "cisco.com"                  (M)
Destination-Realm             [283]     "cisco.com"                  (M)
Destination host name         [293]     "diamclient1.cisco.com"      (M)
Auth-Application-ID           [258]     4                            (M)
Re-Auth-Request-Type          [285]     Authorize-Only               (M)
```

Gy - Sample RAA Packet

```
Session-Id                    [263]     "bng;167;217443434"            (M)
Result-code                   [268]     2001                           (M)
Origin-Realm                  [296]     "cisco.com"                    (M)
Origin-host-name              [264]     "diamclient1.cisco.com"        (M)
Origin-host-name              [264]     "diamclient1.cisco.com"        (M)
```

```
Origin-Realm                   [296]     "cisco.com"                    (M)
```

Gy - CCR-Final Message

```
Session-Id                    [263]   "63;3201010A;4294967295"  (M)
   Origin-host-name           [264]    "cisco123.com"           (M)
   Origin-Realm               [296]    "cisco.com"              (M)
   Auth-Application-ID        [258]    4                        (M)
   CC-request-type            [416]    ccr-final                (M)
   CC-request-number          [415]    3                        (M)
   Service_Context_Id         [461]    "bng1@cisco.com          (M)
   Framed-IP-Address          [8]      "10.0.0.1"    (M)
   User-Name                  [1]      "prepaid-user"           (M)
   Event-Timestamp            [55]     3426644295               (M)
   Termination_Cause          [295]    Diameter Administrative  (M)
   Destination-Realm          [283]    "cisco.com"              (M)
   CC-multiple-service        [456]
      CC-rating-group         [432]    3                        (M)
      Service-Identifier      [439]    5                        (M)
      Used-Service-Unit       [446]
         Vendor, 3GENPP       [10415]
         Reporting-Reason     [872]    FINAL (2)                (M)
         CC-Time              [420]    1                        (M)
      Used-Service-Unit       [446]
         Vendor, 3GENPP       [10415]
         Reporting-Reason     [872]    FINAL (2)                (M)
         CC-Total-Octets      [421]    2                        (M)
         CC-Input-Octets      [412]    1                        (M)
         CC-Output-Octets     [414]    1                        (M)
```

Gy - CCA-Final Message

```
Session-Id                         [263]    "cisco123.com;3201010A;0"  (M)
  Result-code                      [268]    2001                       (M)
  Origin-host-name                 [264]    "cisco123.com"             (M)
  Origin-Realm                     [296]    "cisco.com"                (M)
  CC-request-type                  [416]    ccr-final                  (M)
  CC-request-number                [415]    0                          (M)
  Destination-Realm                [283]    "cisco.com"                (M)
  Auth-Application-ID              [258]    4                          (M)
  CC-session-failover              [418]    NOT_SUPPORTED(0)           (M)
  Credit-Control-Failure-Handling  [427]    CONTINUE (1)               (M)
```