

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 1: 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
Release 6.3.1	Added HTTP Redirect URL with Subscriber Information feature.
Release 6.5.1	These new features were introduced: • HTTP-Redirect Support for Static sessions • HTTP Header Enrichment for BNG Subscribers
Release 6.6.3	Introduced these new features: • Header enrichment on pseduowire headend with subscriber redundancy group support. • HTTP redirect on pseduowire headend with subscriber redundancy group support.
Release 7.0.1	Introduced egress Lawful Intercept (LI) functionality over Pseudowire Headend subscriber interfaces.

Release	Modification
Release 7.3.1	Addded BNG support on Cisco ASR 9000 5th Generation line cards.

The subscriber features covered in this chapter are:

- Access Control List and Access Control List-based Forwarding, on page 2
- Excessive Punt Flow Trap (Deprecated) , on page 6
- Support for Lawful Intercept, on page 6
- TCP MSS adjustment, on page 16
- TCP MSS Adjustment for BNG, on page 20
- Linking to Subscriber Traffic in a Shared Policy Instance Group, on page 23
- Subscriber Session on Ambiguous VLANs , on page 24
- uRPF, on page 29
- Multicast Services, on page 29
- DAPS Support, on page 41
- HTTP Redirect Using PBR, on page 55
- HTTP-Redirect Support for Static Sessions, on page 72
- HTTP Header Enrichment for BNG Subscribers, on page 74
- Idle Timeout for IPoE and PPPoE Sessions, on page 80
- Routing Support on Subscriber Sessions , on page 80
- Traffic Mirroring on Subscriber Session, on page 80
- Randomization of Interim Timeout of Sessions or Services, on page 82
- Controlling Subscriber Plans Using Protocol Options, on page 83
- Additional References, on page 84

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 3.
- Applying the ACL to an access-interface, see Activating ACL, on page 4.



Note ACL is not supported for LAC sessions.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.

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:** `RP/0/RSP0/CPU0:router(config)# 10 permit ipv4 host 9.8.8.9 nexthop 6.6.6.6 7.7.7.7` or	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 ipv6 host 192:2:1:9 nexthop 192:2:6:8`	
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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
```

Excessive Punt Flow Trap (Deprecated)

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Excessive Punt Flow Trap (Feature Deprecated)	Release 7.3.1	This feature is deprecated starting IOS XR Release 7.3.1. To control user or subscriber traffic manually, we recommend you to use Access Control List and Access Control List-based Forwarding, on page 2, Flow Aware QoS, or enable policing through MQC.

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.

[1] PacketCable™ architecture addresses device interoperability and product compliance issues using the PacketCable™ Specifications.

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**.

From Cisco IOS XR Software Release 7.0.1 onwards, the Egress Lawful Intercept feature on BNG supports pseudowire headend (PWHE) subscriber sessions as well

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*.

Maximum supported Lawful Intercept scale depends on the number of Network Processors (NP) on a line card. For example,

- Line cards with one or two NPs: 511 sessions per line card.
- Line cards with more than two NPs: 1024 divided by the number of NPs on the line card.

  You can verify the number of NPs on a line card using the **show controllers np summary all location <location>** command.

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 8.

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.

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 9 and for more information about enabling the mediation device, see Enabling the Mediation Device to Intercept VoIP and Data Sessions, on page 9.

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. Use the **commit** or **end** command.
8. show snmp users
9. show snmp group
10. show snmp view

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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.

	Command or Action	Purpose
		• 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. Most MDs default to or support only MD5.
Step 7	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
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 12.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**aaa intercept**	Enables the radius-based lawful intercept feature.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router(config)# aaa intercept`	**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* } **Example:** `RP/0/RSP0/CPU0:router(config-Dynamic Author)# client 3.0.0.28 vrf default server-key cisco`	Specifies the client with which the AAA server will be communicating. **Note** You can configure the server key in a global mode and also as a per client type key.
Step 7	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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"
```

SNMP Lawful Intercept Using Circuit-Id

Table 3: Feature history table

Feature Name	Release Information	Feature Description
SNMP Lawful Intercept Using Circuit-Id	Release 26.1.1	The SNMP Lawful Intercept using Circuit-Id functionality is now extended to Line Card (LC) based subscribers. Previously, static identification using circuit ID was available only for Route Processor (RP) based subscribers.
SNMP Lawful Intercept Using Circuit-Id	Release 6.2.1	This feature simplifies back-end integration by allowing subscriber traffic interception using a static Circuit-Id. Instead of relying on interface names or session IDs that change with each login, SNMP Lawful Intercept in BNG uses a consistent identifier, making subscriber identification more reliable and seamless.

SNMP lawful intercept (LI) using Circuit-Id in BNG provides a way to intercept the subscriber traffic based on a static identifier rather than the interface name or accounting session ID that changes every time a subscriber logs in to the network. This feature reduces the constraints on the back end, thereby providing a seamless integration.

For more details on lawful intercept, refer *System Security Configuration Guide for Cisco ASR 9000 Series Routers.*

For more details on implementing SNMP, refer *System Management Configuration Guide for Cisco ASR 9000 Series Routers.*

Configuration Example

Execute these commands on SNMP server:

- Create lawful intercept-mediation device (LI-MD):

```
# Create LI-MD
#  CDID : 11
```

```
#  MD IP: 5.5.5.2
#  UDP PORT: 1234
#  TIME TILL: 2016 (07-E0), FEB (02), 29 (1D)

snmpset -v3 -u li-user -A cisco123 -l authNoPriv 25.25.25.1 \
   1.3.6.1.4.1.9.9.399.1.1.2.1.13.11 i 5 \
   1.3.6.1.4.1.9.9.399.1.1.2.1.2.11 i 1  \
   1.3.6.1.4.1.9.9.399.1.1.2.1.3.11 x "05 05 05 02" \
   1.3.6.1.4.1.9.9.399.1.1.2.1.4.11 u 1234 \
   1.3.6.1.4.1.9.9.399.1.1.2.1.5.11 i 0 \
   1.3.6.1.4.1.9.9.399.1.1.2.1.11.11 i 1 \
   1.3.6.1.4.1.9.9.399.1.1.2.1.10.11 x "07 E0 02 1D 00 00 00 00 2D 00 00" \
   1.3.6.1.4.1.9.9.399.1.1.2.1.13.11 i 4
```

- Create subscriber TAP stream:

```
# CREATE SUBSCRIBER TAP STREAM BASED ON CID
#  STREAM INDEX: 22
#  SUBS CID: cid101

snmpset -v3 -u li-user -A cisco123 -l authNoPriv 25.25.25.1 \
   1.3.6.1.4.1.9.9.400.1.1.2.1.3.11.22 s "cid101" \
   1.3.6.1.4.1.9.9.400.1.1.2.1.2.11.22 i 4
```

- Enable TAP:

```
# ENABLE TAP ON THE STREAM
#   MD CCDI: 11
#   TAP STREAM ID: 22
#   STREAM TYPE: 3 (SUBS)

snmpset -v3 -u li-user -A cisco123 -l authNoPriv 25.25.25.1 \
  1.3.6.1.4.1.9.9.399.1.2.1.1.2.11.22 i 3 \
  1.3.6.1.4.1.9.9.399.1.2.1.1.3.11.22 i 1 \
  1.3.6.1.4.1.9.9.399.1.2.1.1.6.11.22 i 4
```

Verification

Execute this command on SNMP server to verify the statistics:

```
snmpwalk -v3 -u li-user -A cisco123 -l authNoPriv 25.25.25.1 1.3.6.1.4.1.9.9.399
```

```
SNMPv2-SMI::enterprises.9.9.399.1.1.1.0 = INTEGER: 54
SNMPv2-SMI::enterprises.9.9.399.1.1.2.1.2.11 = INTEGER: 1
SNMPv2-SMI::enterprises.9.9.399.1.1.2.1.3.11 = Hex-STRING: 05 05 05 02
SNMPv2-SMI::enterprises.9.9.399.1.1.2.1.4.11 = Gauge32: 1234
SNMPv2-SMI::enterprises.9.9.399.1.1.2.1.5.11 = INTEGER: 0
SNMPv2-SMI::enterprises.9.9.399.1.1.2.1.7.11 = INTEGER: 34
SNMPv2-SMI::enterprises.9.9.399.1.1.2.1.10.11 = Hex-STRING: 07 E0 02 1D 00 00 00 00 2D 00
00
SNMPv2-SMI::enterprises.9.9.399.1.1.2.1.11.11 = INTEGER: 1
SNMPv2-SMI::enterprises.9.9.399.1.1.2.1.12.11 = INTEGER: 1
SNMPv2-SMI::enterprises.9.9.399.1.1.2.1.13.11 = INTEGER: 1
SNMPv2-SMI::enterprises.9.9.399.1.1.3.0 = Hex-STRING: A0
SNMPv2-SMI::enterprises.9.9.399.1.2.1.1.2.11.22 = INTEGER: 3
SNMPv2-SMI::enterprises.9.9.399.1.2.1.1.3.11.22 = INTEGER: 1
SNMPv2-SMI::enterprises.9.9.399.1.2.1.1.4.11.22 = Counter32: 13921356
SNMPv2-SMI::enterprises.9.9.399.1.2.1.1.5.11.22 = Counter32: 0
SNMPv2-SMI::enterprises.9.9.399.1.2.1.1.6.11.22 = INTEGER: 1
```

```
SNMPv2-SMI::enterprises.9.9.399.1.3.1.0 = INTEGER: 600
SNMPv2-SMI::enterprises.9.9.399.1.3.2.0 = INTEGER: 10
```

TCP MSS adjustment

The TCP Maximum Segment Size (MSS) adjustment is a network capability that

- intercepts TCP SYN packets and scans for the first option after the TCP header
- tunes the MSS value to the configured value and updates the checksum accordingly, and
- ensures compatibility across various interface types, including physical, subinterfaces, bundle interfaces, and tunnel interfaces, for both ingress and egress traffic.

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
TCP MSS adjustment	Release 25.2.1	You can now enhance network performance by optimizing TCP packet sizes, reducing fragmentation, and improving data transmission efficiency. This capability ensures compatibility and performance optimization in scenarios where mismatched MSS values between endpoints could lead to suboptimal performance or packet loss. By dynamically tuning the MSS in TCP SYN packets, the feature ensures seamless communication. In its default operation, the feature applies the default MSS value consistently across all interfaces when no custom MSS value is configured. For custom configurations, it matches the configured MSS value, ensuring optimal data flow and preventing packet loss. This functionality is critical for maintaining robust network performance and addressing challenges in data transmission. The default value of TCP MSS is 1436 bytes. The supported range for TCP MSS is 1280 to 1535 bytes. This feature is supported on: • Cisco ASR 9000 High Density 100GE Ethernet line cards The feature introduces these commands: • **ipv4 tcp-mss-adjust enable** • **ipv6 tcp-mss-adjust enable** • **hw-module tcp-mss-adjust**

Challenges of TCP packet fragmentation in networks with varying MTU sizes

TCP packets may encounter fragmentation when traversing networks with varying MTU sizes, especially in scenarios like PPPoE, where additional headers increase packet size beyond link limits. For example, a client's advertised TCP MSS, combined with IP, TCP, and PPPoE headers, can exceed the MTU, causing inefficiencies and higher latency.

A solution to prevent packet fragmentation and optimize network performance

To combat TCP fragmentation, the TCP MSS adjustment feature intercepts TCP SYN packets, scans for the MSS option in the TCP header, and readjusts the MSS value to a configured value.. By intercepting these packets, the feature ensures that the advertised MSS aligns with the smallest MTU along the packet's path, accounting for encapsulations like PPPoE. This proactive tuning prevents packet fragmentation, optimizing network performance and ensuring seamless data transmission across diverse network environments. This capability is crucial for maintaining network efficiency and reliability, especially in scenarios with varying MTU sizes.

How TCP MSS adjustment works

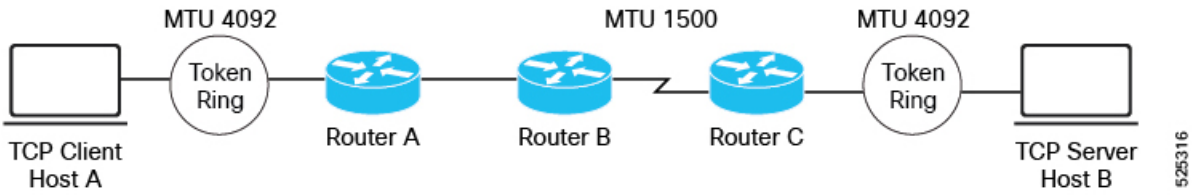
Summary

The TCP MSS adjustment feature dynamically modifies the MSS in TCP SYN packets to prevent fragmentation in networks with varying MTU sizes. The key components involved in this process are:

- TCP Client (Host A): Initiates TCP connections and is connected to a Token Ring with an MTU of 4092.
- Routers (A, B, C): Form the network path, with Router B having an MTU of 1500.
- TCP Server (Host B): Receives TCP connections and is connected to a Token Ring with an MTU of 4092.

Workflow

Figure 1: TCP MSS Adjustment



The TCP MSS adjustment process involves the following stages:

1. Initial TCP SYN packet: The TCP client (Host A) sends a TCP SYN packet to the TCP server (Host B), including the MSS option.
2. MTU change and potential fragmentation: As the packet traverses the network, it encounters a link with a smaller MTU, for example, 1500 bytes, on the link between Router A and Router B). Without MSS adjustment, packets exceeding this MTU would be fragmented.
3. TCP MSS interception: The router intercepts the TCP SYN packet and scans the TCP header options for the MSS value.
4. MSS value adjustment: The router adjusts the MSS value to a configured value suitable for the smallest MTU along the path, ensuring that TCP segments do not exceed the MTU of any link.
5. Checksum update: The router updates the TCP checksum to reflect the MSS value change.
6. Forwarding the modified SYN packet: The router forwards the modified TCP SYN packet to the server.

7. Subsequent TCP communication: The TCP server uses the adjusted MSS value for subsequent communication, ensuring that TCP segments remain within MTU limits and avoid fragmentation.

   The TCP MSS adjustment feature ensures efficient and fragmentation-free TCP communication across networks with varying MTU sizes.

Guidelines for TCP MSS adjustment

Key considerations and best practices for ensuring effective implementation of TCP MSS adjustment include:

- Interface compatibility: The feature is applicable to all interface types, ensuring seamless operation across physical, sub-interfaces, bundle interfaces, and tunnel interfaces.
- Protocol support: The feature supports both IPv4 and IPv6 traffic, even if only one protocol is configured.
- Default and custom values: The feature operates with default MSS values and allows adjustments as per configuration requirements. The default value of TCP MSS adjustment is 1436 bytes. The supported range for TCP MSS adjustment is 1280 to 1535 bytes.
- Hardware support
  - Cisco ASR 9000 High Density 100GE Ethernet line cards
- You can configure the hardware module settings multiple times. When a new configuration is applied, the router evaluates and prioritizes the configurations based on a predefined order of precedence. The precedence categories are as follows:
  1. Location-Specific NP Specific: Applies to a specific node in the router associated with a specific NPU. Example: **`location 0/0/CPU0 NP 1`**
  2. Location-Specific NP All: Applies to a specific node in the router associated with all NPUs. Example: **`location 0/0/CPU0 NP All`**
  3. Location All NP Specific: Applies to all nodes in the router associated with a specific NPU. Example: **`location All NP 1`**
  4. Location All NP All: Applies to all nodes in the router associated with all NPUs. Example: **`location All NP All`**

  For example, consider configuring the TCP MSS adjustment value using the **`hw-module tcp-mss-adjust`** command:

  1. If you configure **`location 0/0/CPU0 NP All`** first, followed by **`location All NP All`**:
     - The TCP MSS value for **`location 0/0/CPU0 NP All`** will take precedence for the node at **`0/0/CPU0`**. This means the node at **`0/0/CPU0`** will use the TCP MSS value defined for **`location 0/0/CPU0 NP All`**.
     - For all other nodes in the router, the TCP MSS value defined for **`location All NP All`** will apply.
  2. If you configure **`location All NP All`** first, followed by location **`0/0/CPU0 NP All`**:
     - The TCP MSS value for **`location 0/0/CPU0 NP All`** will still take precedence for the node at **`0/0/CPU0`**.

- In this case, the setting for **location 0/0/CPU0 NP All** will override the broader configuration for **location All NP All** specifically for this node.
- For all other nodes, the TCP MSS value for **location All NP All** will apply, provided no other specific configurations have been made.

Configure TCP MSS adjustment

Configure TCP MSS for TCP connections passing through the router.

Before you begin

Follow these steps to ensure the router is ready for configuration:

- Ensure that the router is operational and accessible.
- Verify that the required interface configurations are in place.

Procedure

Step 1 Set global TCP MSS values.

Example:

```
Router(config)# hw-module location 0/0/CPU0 tcp-mss-adjust np 1 value 1280
Router(config)# hw-module location 0/1/CPU0 tcp-mss-adjust np 1 value 1280
```

Step 2 Enable TCP MSS.

The example shows the configuration of TCP MSS on a bundle subinterface.

Example:

```
Router(config)# interface Bundle-Ether1.602
Router(config-subif)# ipv4 address 10.193.240.49 255.255.255.240
Router(config-subif)# ipv4 tcp-mss-adjust enable
Router(config-subif)# encapsulation dot1q 602
Router(config-subif)# commit
```

The example shows the configuration of TCP MSS on a physical subinterface.

Example:

```
Router(config)# interface TenGigE0/0/1/0.1
Router(config-subif)# ipv4 address 192.168.140.21 255.255.255.252
Router(config-subif)# ipv4 tcp-mss-adjust enable
Router(config-subif)# encapsulation dot1q 991
Router(config-subif)# commit
```

Step 3 Set global TCP MSS values.

Example:

```
Router(config)# hw-module location 0/0/CPU0 tcp-mss-adjust np 1 value 1280
Router(config)# hw-module location 0/1/CPU0 tcp-mss-adjust np 1 value 1280
```

TCP MSS adjustment is configured successfully on the specified interfaces and globally on the router.

Step 4 Running configuration.

Example:

```
Router(config)# show running config interface Bundle-Ether1.602
Bundle-Ether1.602
  ipv4 address 10.193.240.49 255.255.255.240
  ipv4 tcp-mss-adjust enable
  encapsulation dot1q 602
!

Router(config)# show running config interface TenGigE0/0/1/0.1
interface TenGigE0/0/1/0.1
  ipv4 address 192.168.140.21 255.255.255.252
  ipv4 tcp-mss-adjust enable
  encapsulation dot1q 991
!
```

TCP MSS Adjustment for BNG

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 2: 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 21.

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. Use the **commit** or **end** command.

5. **configure**
6. **vpdn**
7. **l2tp tcp-mss-adjust** *max-segment-size*
8. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 5	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 6	**vpdn** **Example:**	Enables the vpdn configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# vpdn`	
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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).

For more information about this feature, see the Configuring Subscriber Features chapter in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide*.

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 24.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	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 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:
	`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`
		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 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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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**.

For more information about the Outer-VLAN range feature, see the *Configuring Subscriber Features* chapter in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide*. For complete command reference of the Outer-VLAN range specific commands, see the *BNG Interfaces Commands* chapter in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*.

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

Procedure

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.

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 30.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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.
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions:

	Command or Action	Purpose
		• **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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 31.

To specify the mode for Multicast HQoS, see Configuring Multicast HQOS Correlation Mode or Passive Mode, on page 33.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions:

	Command or Action	Purpose
		• **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

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

	Command or Action	Purpose
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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 34. For more information about configuring the IGMP for subscriber interfaces, see Configuring IGMP Parameters for Subscriber Interfaces, on page 38.

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 36.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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:** `RP/0/RSP0/CPU0:router(config-igmp)# vrf vrf1`	Enters the VRF configuration mode.
Step 7	**traffic profile** *profile-name* **Example:**	Configures the route-policy to be used to map the bandwidth profile.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-igmp-vrf1)# traffic profile routepolicy1`	
Step 8	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use the **commit** or **end** command.
6. **show igmp unicast-qos-adjust statistics**
7. **show igmp unicast-qos-adjust statistics interface** *interface-name*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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:** `RP/0/RSP0/CPU0:router(config-igmp)# vrf vrf1`	Enters the vrf configuration mode.
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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:**	Enters the ppp type mode to configure igmp for subscriber interfaces.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-dynamic-template)# type ppp foo`	
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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 QoS Correlation for IPoE Subscribers

From Cisco IOS XR Software Release 6.2.1and later, the IGMP QoS correlation feature is extended to IPoE subscribers, in addition to PPPoE subscribers.

For more information about this feature, see the *Configuring Subscriber Features* chapter in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide*. For complete command reference, see the *Multicast Commands* chapter in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*.

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 40. 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. Use the **commit** or **end** command.
5. **show igmp interface**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
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 42. To create a configuration pool for IPv6, see Creating a Configuration Pool Submode, on page 43.

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 45. You can specify parameters such as a range of IPv6 addresses. For more information, see Specifying a Range of IPv6 Addresses, on page 46. To specify a utilization threshold, see Specifying a Utilization Threshold, on page 48. To specify a set of prefixes or addresses inside a subnet, see Specifying a Set of Addresses or Prefixes Inside a Subnet, on page 51. You can also specify the length of a prefix. For more information, see Specifying the Length of the Prefix, on page 49.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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.

	Command or Action	Purpose
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use the **commit** or **end** command.
4. **configure**
5. **pool vrf** *vrf_name* **ipv6** *ipv6-pool-name*
6. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 4	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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.

	Command or Action	Purpose
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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.

	Command or Action	Purpose
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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.

	Command or Action	Purpose
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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
```

Dynamic Modification of DAPS Pool

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
Dynamic Modification of DAPS Pool	Release 7.4.1Release 6.8.1	A Distributed Address Pool Service (DAPS) provides dynamic IP address allocation to the BNG subscriber. This feature allows you to modify the DAPS pool configuration in live networks without bringing down the interface or the BNG subscriber session. You can dynamically release the IP address, and thereby release the active subscriber session. This functionality in turn helps you to configure a new IP address pool with the same IP address range or a different one that can be used for a new subscriber. This feature introduces the **pool onfly pppoe** command.

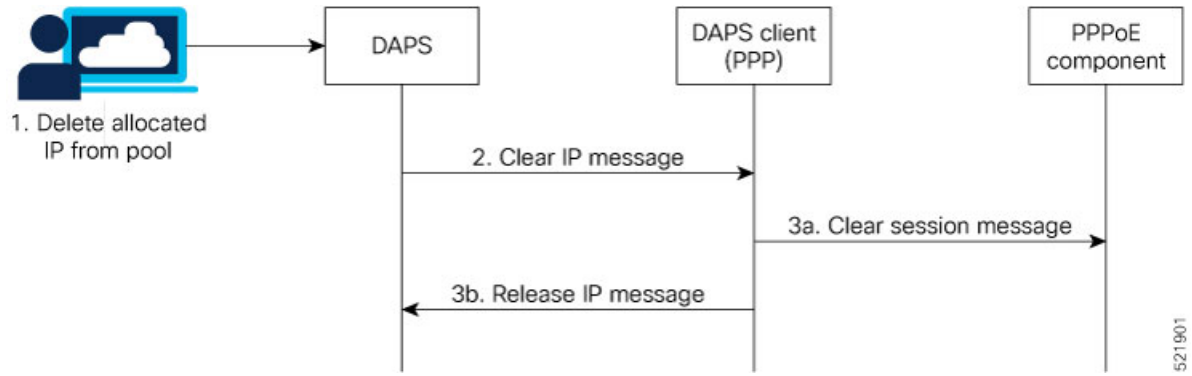
A Distributed Address Pool Service (DAPS) supports dynamic IP address allocation to the BNG subscriber. With the introduction of dynamic modification of DAPS pool feature, you can modify the IP address pool configuration on the fly without bringing down the interface or the subscriber session.

The following is the expected behavior when you dynamically modify the DAPS pool configuration:

- If you reduce the IP address range, and if the number of subscriber sessions is more than the remaining IP addresses in the pool, then the system releases those additional subscribers to match the remaining numbers in the pool.
- If you increase the IP address range, then there is no impact to the already allocated IP addresses.
- If you remove the pool configuration altogether, then the system releases all the allocated IP addresses.

Call Flow of Dynamic Modification of DAPS Pool

Figure 3: Call Flow of Dynamic Modification of DAPS Pool



The call flow of dynamic modification of DAPS pool involves these steps within the router components:

1. The network administrator edits the DAPS pool configuration on BNG router and deletes an IP address from the address pool.
2. The DAPS sends a *CLEAR IP* message to the DAPS client (PPP).
3. **a:** The DAPS client in turn sends a *CLEAR SESSION* message to the PPPoE component to bring down the subscriber session. The PPPoE component then clears the sessions, and returns a *RELEASE IP* message to the DAPS client.

   b: The DAPS client forwards this *RELEASE IP* message to DAPS, and DAPS releases that IP address.

   Meanwhile, the PPPoE component continues with the subscriber session deletion process, and the session eventually gets deleted.

Restrictions

These restrictions apply to dynamic modification of DAPS pool feature:

- For the subscriber session to be released after the configuration changes are done, you may have to wait for approximately 5 to 20 minutes, depending on the number of IP addresses allocated from an address pool.
- The support is for PPPoE sessions; not for IPoE sessions.
- Subscriber redundancy group (SRG) is not supported.

How to Enable Dynamic Modification of DAPS Pool

To enable dynamic modification of DAPS pool, use the **pool onfly pppoe** command in Global Configuration mode.

Configuration Example

```
Router#configure
Router(config)#pool onfly pppoe
Router(config)#commit
```

Running Configuration

```
pool onfly pppoe
!
```

Verification

This command output shows the current IP address pool configuration:

```
Router#show run pool
pool vrf default ipv4 testv4
address-range 192.168.10.10 192.168.10.110
!
```

This command output displays the pool statistics:

```
Router#show pool statistics
Mon Jun 14 11:16:23.306 UTC
Debug stats:
alloc_req_msg_recv: 1
alloc_rsp_msg_send: 1
shadow_ntfy_msg_recv: 5
shadow_replay_req_msg_send: 1
shadow_replay_req_msg_recv: 2
shadow_replay_end_msg_send: 2
shadow_replay_end_msg_recv: 1
shadow_upload_req_msg_send: 1
shadow_upload_req_msg_recv: 1
shadow_purge_msg_recv: 1
proxy_resync_complete_msg_send: 5
shadow_alloc_create_recv: 5
```

Associated Commands

- **pool onfly pppoe**

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.

From Cisco IOS XR Software Release 6.6.3 onwards, the HTTP redirect feature on BNG is enhanced to support pseudowire headend (PWHE) subscriber sessions as well. This enhancement is available along with subscriber redundancy group (SRG) support, thereby providing geographical redundancy for such sessions.

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 56.
- 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 60.
- 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 62.
- Creating the dynamic template to apply the service policy. See Configuring Dynamic Template for Applying HTTPR Policy, on page 64.

To configure a web logon that specifies a time limit to perform the authentication, see Configuring Web Logon, on page 66.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.

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**]	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.

	Command or Action	Purpose

	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`
		• 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 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
		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

	Command or Action	Purpose
	`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`	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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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 56 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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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.

	Command or Action	Purpose
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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 56 and Configuring Class Maps for HTTP Redirection, on page 60 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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router# 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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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 62 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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

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. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**class-map type control subscriber match-all** *classmap_name* **Example:**	Configures a subscriber control class-map with the match-all match criteria.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# class-map type control subscriber match-all IP_UNAUTH_COND`	
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.
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.

	Command or Action	Purpose
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.
Step 18	*sequence_number* **disconnect** **Example:** `RP/0/RSP0/CPU0:router(config-pmap-c)# 10 disconnect`	Disconnects the session.
Step 19	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes.

	Command or Action	Purpose
		• **Cancel** —Remains in the configuration session, without committing the configuration changes.

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
```

HTTP Redirect URL with Subscriber Information

The existing HTTP Redirect feature in BNG is enhanced to provide subscriber information in addition to the redirect portal address in the redirect URL. This information includes the user name of the subscriber (in a Cisco Attribute-Value-Pair named *parsed-user-name*), the IP address of the subscriber and the source-identifier of the BNG. This functionality provides a mechanism to authenticate the subscriber as well as to enhance the web server, which in turn can be used in multiple use cases of redirect scenarios. One such use case is, providing a subscriber-specific web page to the subscriber in a redirect scenario. This feature works with both RADIUS and DIAMETER protocols.

For more information about this feature, see the *Configuring Subscriber Features* chapter in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide*. For complete command reference, see the *Control Policy Commands* chapter in *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*.

Use Cases of HTTP Redirect URL with Subscriber Information

In general, HTTP redirect feature can be activated dynamically or in authorization or authentication failure scenarios. This section covers various redirect scenarios for the subscriber and the expected feature behavior of HTTP redirect URL with subscriber information.

1. **Redirect triggered by DIAMETER GX Re-Auth-Request (RAR) or RADIUS Change-of-Authorization (CoA) message**

   This is a use case of dynamically activating HTTP redirect feature which is observed in scenarios such as:

   - A DSL subscriber who is present in the AAA database with `Treatment Status` as `redirect`.
   - A Fiber to the home (FTTH) subscriber who is present in the AAA database, but without having a privileged user profile, and with 'Treatment Status' as 'redirect'.

   Once the subscriber session is created by the BNG router, it sends a Credit-Control-Request-Initial (CCR-I) message to the Policy and charging rule function (PCRF) server, which is the DIAMETER server. In response to this, the PCRF server sends the Credit-Control-Answer-Initial (CCA-I) message to the BNG, along with the policy to be applied for the subscriber. Later, for example, when the user exceeds the usage quota, the PCRF server sends a RAR message whereby the redirect policies are applied dynamically for the subscriber.

2. **Redirect triggered by GX Credit-Control-Answer Initial (CCA-I) message during the bring-up of the subscriber session**

   Here, the CCR-I and CCA-I message exchange happens as mentioned in use case 1 above. The redirect service policy for the subscriber which is carried in the CCA-I message is then downloaded by the BNG and applied to the subscriber profile. Whenever the subscriber tries to access a web portal, it is redirected by the BNG to the respective web server as per the redirect URL configured for that subscriber. The communication then happens between the subscriber and the redirect web server.

3. **Redirect in Access-Accept message scenario**

   The redirect URL with subscriber information feature is enabled if the Access-Accept message sent from the AAA server to the BNG contains the Cisco AVP **parsed-user-name** as part of the user profile.

4. **Redirect in authentication failure scenarios**

   The redirect URL with subscriber information feature is not enabled in access reject scenarios such as:

   - A subscriber who is not provisioned in the AAA database
   - A DSL subscriber with incorrect DHCP Option 82
   - An FTTH subscriber with incorrect DHCP Option 82

   The subscriber username, subscriber IP address and BNG source-identifier are not present in these scenarios. The AAA server sends the Access-Reject message without the *parsed-user-name* attribute to the BNG which results in subscriber authentication failure in the above scenarios. A default string with value as `NOTAVAILABLE` is sent in the Access-Reject message in such scenarios. The BNG redirects such subscribers to a portal server using a pre-configured `authentication-failed-redirect` URL. The subscriber accounting can be decided by enabling or disabling **accounting** on the BNG.

Configure HTTP Redirect URL with Subscriber Information

The configuration steps remain the same for redirect URL with and without subscriber information. For detailed steps, see HTTP Redirect Using PBR, on page 55.

Running Configuration

```
/* Access List Configuration */
ipv4 access-list httpr-acl-example
 10 permit tcp any any eq www syn
 20 permit tcp any any eq www ack
 30 permit tcp any any eq www
!

/* Class Map Configuration */
class-map type traffic match-any httpr-class-example
 match access-group ipv4 httpr-acl-example
 end-class-map
!
policy-map type pbr httpr-policy-example
 class type traffic httpr-class-example
  http-redirect http://example.com/index.php
 !
class type traffic class-default
 !
 end-policy-map
!

/* Service Policy Configuration */
dynamic-template
 type service httpr-service-example
  service-policy type pbr httpr-policy-example
 !
!
```

Verification

Use this command to verify if the subscriber policy is applied or not:

```
Router# show policy-map type pbr interface Bundle-Ether1.1
Sun Feb 19 22:19:32.289 EDT
Bundle-Ether1.1 input: httpr-policy-example
Policy Name: httpr-policy-example
Class httpr-class-example
  Classification statistics             (packets)
    Matched             :                  26
  Httpr statistics                      (packets)
    Requests Received   :                  26
    Responses Sent      :                   0
    Redirect drops      :                  26
Class class-default
  Classification statistics          (packets/bytes) (May be 10secs old)
    Matched             :                 10/940
```

Associated Commands

- class-map
- dynamic-template
- http-redirect
- policy-map

- show policy-map

HTTP-Redirect Support for Static Sessions

From Cisco IOS XR Software Release 6.5.1 onwards, BNG extends the HTTP redirect feature support for static sessions.

For a detailed information on HTTP Redirect feature, refer to the section HTTP Redirect Using PBR, on page 55.

Restrictions for HTTP Redirect for Static Sessions

HTTP redirect for static sessions in BNG is subjected to the below restrictions:

- The maximum size of the redirect URL cannot be greater than 256 bytes (or characters). The redirect URL includes both the original information, such as the portal webserver or the IP address information, as well as the extension information which is packed in a defined format.

Configure HTTP Redirect for Static Sessions

Configuration Steps

The below section shows the steps to configure a redirect policy in which an authentication failure results in the subscriber being redirected to a portal server.

For instructions on creating access lists that define the redirected and open-garden permissions. See, Identifying HTTP Destinations for Redirection, on page 56.

For instructions on 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 60.

```
Router# config terminal
Router(config)# ipv4 access-list httpr-acl-cfg
Router(config-ipv4-acl)# 10 permit tcp any any eq www syn
Router(config-ipv4-acl)# 20 permit tcp any any eq www ack
Router(config-ipv4-acl)# 30 permit tcp any any eq www
Router(config-ipv4-acl)# exit
Router(config)# class-map type traffic match-any httpr-class-cfg
Router(config-cmap)# match access-group ipv4 httpr-acl-cfg
Router(config-cmap)# end-class-map
Router(config)# policy-map type pbr httpr-policy-cfg
Router(config-pmap)# class type traffic httpr-class-cfg
Router(config-pmap-c)# http-redirect http://203.0.113.103/index.php
Router(config-pmap-c)# exit
Router(config-pmap)# class type traffic class-default
Router(config-pmap-c)# drop
Router(config-pmap-c)# exit
Router(config-pmap)# end-policy-map
Router(config)# dynamic-template
Router(config-dynamic-template)# type ipsubscriber httpr-service-cfg
Router(config-dynamic-template-type)# service-policy type pbr httpr-policy-cfg
Router(config-dynamic-template-type)# exit
Router(config-dynamic-template)# exit
Router(config)# policy-map type control subscriber POLICY1
Router(config-pmap)# service-policy type pbr httpr-policy-cfg
```

```
Router(config-pmap-e)# event authentication-failure match-first
Router(config-pmap-e)# class type control subscriber CL1 do-until-failure
Router(config-pmap-c)# 2 activate dynamic-template httpr-service-cfg
Router(config-pmap-c)# exit
Router(config-pmap-e)# exit
Router(config-pmap)# end-policy-map
Router(config)# interface gigabitEthernet 0/0/0/0
Router(config-if)# bundle id 1 mode on
Router(config-if)# exit
Router(config)# interface bundle-ether 1
Router(config-if)# mac-address a1.b1.c1
Router(config-if)# exit
Router(config)# interface bundle-ether 1.100
Router(config-subif)# ipv4 address 192.0.2.10 255.255.255.0
Router(config-subif)# ipv6 address 2001:DB8::10/32
Router(config-subif)# ipv6 nd dad attempts 0
Router(config-subif)# encapsulation dot1q 100
Router(config-subif)# service-policy type control subscriber POLICY1
Router(config-subif)# ipsubscriber interface
Router(config-subif)# commit
```

Running Configuration

The below example shows the running configuration for a redirect policy in which an authentication failure results in the subscriber being redirected to a portal server.

```
ipv4 access-list httpr-acl-cfg
 10 permit tcp any any eq www syn
 20 permit tcp any any eq www ack
 30 permit tcp any any eq www
!
class-map type traffic match-any httpr-class-cfg
 match access-group ipv4 httpr-acl-cfg
 end-class-map
!
policy-map type pbr httpr-policy-cfg
 class type traffic httpr-class-cfg
  http-redirect http://203.0.113.103/index.php
 !
 class type traffic class-default
  drop
 !
 end-policy-map
!
dynamic-template
 type ipsubscriber httpr-service-cfg
  service-policy type pbr httpr-policy-cfg
 !
!
policy-map type control subscriber POLICY1
 event authentication-failure match-first
  class type control subscriber CL1 do-until-failure
   2 activate dynamic-template httpr-service-cfg
end-policy-map
!
interface GigabitEthernet0/0/0/0
 bundle id 1 mode on
!
interface Bundle-Ether 1
 mac-address a1.b1.c1
!
interface Bundle-Ether 1.100
```

```
ipv4 address 192.0.2.10 255.255.255.0
ipv6 address 2001:DB8::10/32
ipv6 nd dad attempts 0
encapsulation dot1q 100
service-policy type control subscriber POLICY1
ipsubscriber interface
```

HTTP Header Enrichment for BNG Subscribers

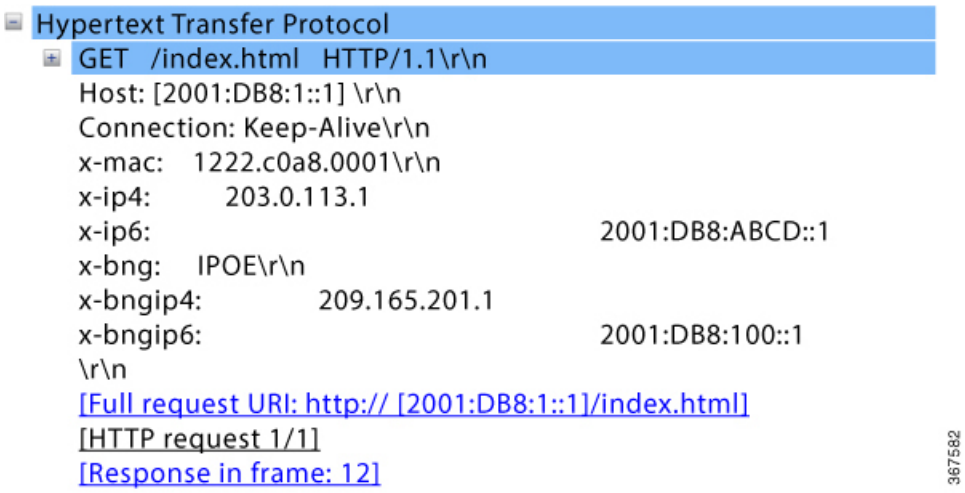
Table 6: Feature History Table

Feature Name	Release Information	Feature Description
HTTP Header Enrichment for BNG Subscribers	Release 7.3.1	The feature redirectes HTTP GET request packets sent by a broadband subscriber back to the original destination server with enriched header parameters. From this release onwards, a place-holder value is retained instead of the actual enriched parameters, thereby preventing any accidental sharing of BNG subscriber information with the subscribers.

This feature enables you to intercept the HTTP GET request packet sent by a broadband subscriber, add custom HTTP headers to it and then forward the enriched HTTP GET request to the original destination server. This provides the capability to add information within the HTTP header, such as subscriber and BNG information, which the server can decode in order to decide whether to allow or deny the request.

From Cisco IOS XR Software Release 6.6.3 onwards, the HTTP header enrichment feature on BNG is enhanced to support pseudowire headend (PWHE) subscriber sessions as well. This enhancement is available along with subscriber redundancy group (SRG) support, thereby providing geographical redundancy for such sessions.

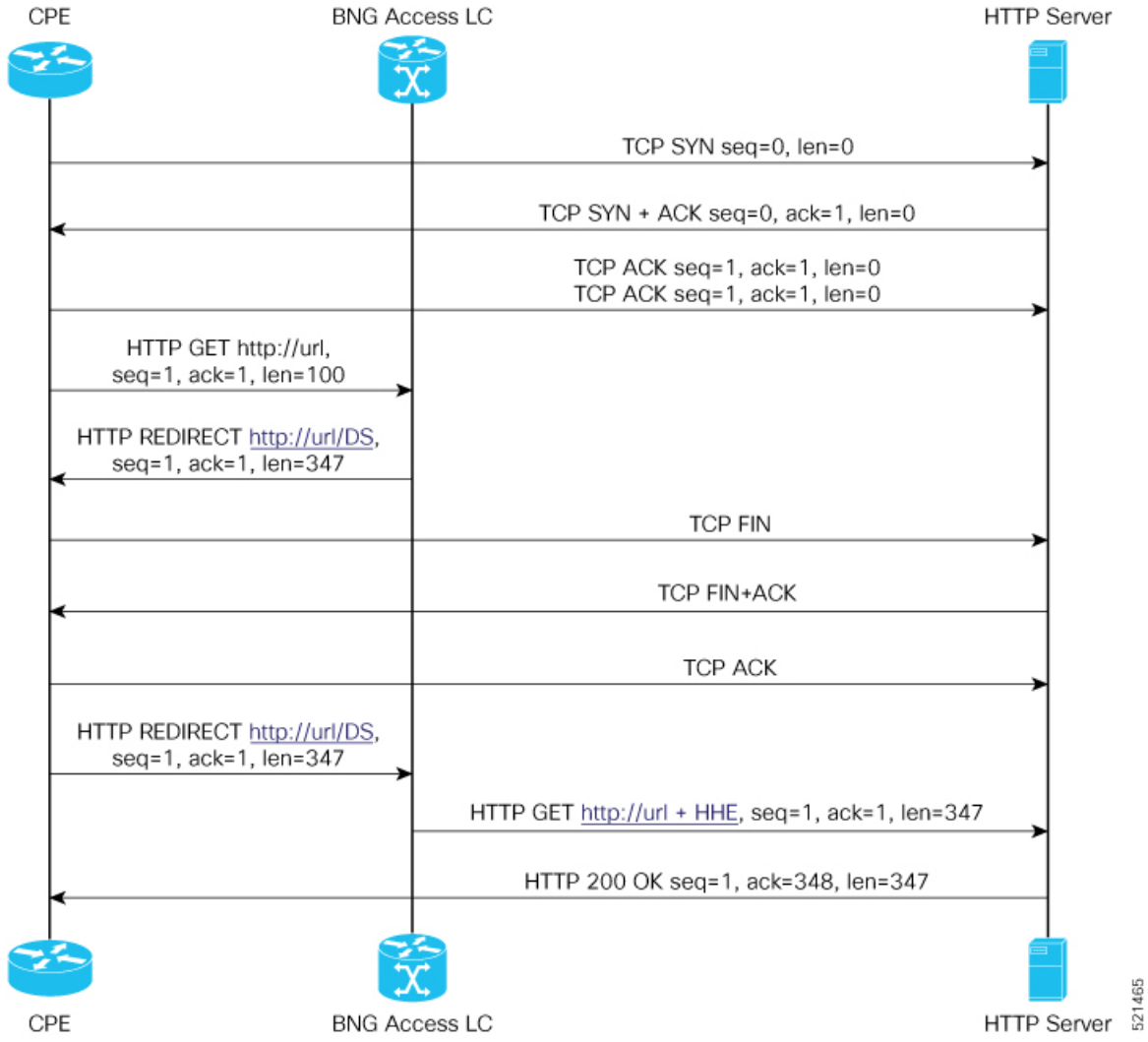
Figure 4: Example of an enriched HTTP header



Starting with Cisco IOS XR Software Release 7.3.1, an intercepted HTTP GET packet is redirected back to HTTP client with enriched header field, but instead of actual value of the enriched parameters, only place-holder value is retained. This method prevents leakage of BNG subscriber information back to the subscriber. Once the new HTTP GET packet is re-intercepted, the actual value is filled instead of place-holder value and then forwarded to HTTP server. Furthermore, sequence number synchronization issue is also avoided between client and server because the HTTP client and server are in sync with actual HTTP packet length. The following events take place:

1. CPE sends HTTP get request to BNG.
2. BNG redirects it to CPE with a dummy signature.
3. CPE resends HTTP get with dummy signature.
4. BNG intercepts the HTTP GET request packet, adds additional parameters to it and then forwards the enriched HTTP GET request to the original destination server.

Figure 5: Sequence of events for HTTP Header Enrichment



Note The number of supported BNG subscribers per Line Card has been increased to 128K from 64K, beginning with Cisco IOS XR Software Release 7.3.1.

Restrictions for HTTP Header Enrichment

HTTP header enrichment in BNG is subjected to the below restrictions:

- An HTTP request or response of length greater than 2000 bytes cannot be processed.
- This feature is not supported in cases where the HTTP request or response is fragmented.
- Only the first HTTP request packet from the client is enriched. After a SUCCESS response is obtained, no further HTTP packets will be received from subscriber for the entire duration of the timeout period, which is around 8 hours.

- The state of the BNG identifier interface will not be considered. In other words, even if the BNG identifier interface is shutdown the configured IP address will be used for enrichment.
- There can be additional spaces in encoded x-headers.
- IP options are not supported with HTTP header enrichment.
- HTTPS enrichment is not supported.
- The HTTP server or proxy need to be using TCP connection.
- In the case of high availability scenarios, the in-progress HTTP transactions will be dropped and the client will have to retry.
- HTTP header enrichment based on destination URL is not supported.
- This feature is supported only for IPoE and PWHE sessions.
- In the scenario where the hostname of the BNG is changed, the new hostname is used for both the new and exisiting subscribers. However, randomly changing the hostname of the router when there are exisiting subscribers present, is not recommended.
- This feature does not support random changes in the subscriber attributes such as IP address and MAC address.
- If the BNG interface is changed, the HTTP header enrichment feature uses the IP address of the new BNG interface for existing and new subscribers. However, randomly changing the BNG interface when existing subscribers are present on the router, is not recommended.
- In the scenario where the BNG interface is not configured, the HTTP header enrichment features uses 0.0.0.0 as the IPv4 address and 0::0 as the IPv6 address. This is a misconfiguration and is not recommended.
- It is not recommended to configure extra parameters in the **http-enrichment parameter-list** that is not mentioned in **subscriber http-enrichment parameter-list**. This is a misconfiguration and results in incorrect encoding.

Configure HTTP Header Enrichment for BNG Subscribers

Configuration Steps

The below section shows how to configure a PBR based policy-map with http-enrichment enabled on one of the classes:

```
Router# config terminal
Router(config)# policy-map type pbr http-enrichment-policy
Router(config-pmap)# class type traffic open-garden-class
Router(config-pmap-c)# transmit
Router(config-pmap-c)# exit
Router(config-pmap)# class type traffic http-enrich-class1
Router(config-pmap-c)# http-enrichment parameter-list subscriber-mac hostname bng-interface
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# drop
Router(config-pmap-c)# commit
```

The configuration steps for configuring a primary-list of http-enrichment parameters that are used in http-enrichment actions configured across all class-maps is shown below:

```
Router(config)# subscriber
Router(config-subscriber)# http-enrichment parameter-list subscriber-mac hostname
bng-interface
Router(config-subscriber)# commit
```

The configuration steps to set the bng identifier interface is shown below:

```
Router(config)# subscriber
Router(config-subscriber)# bng-interface Loopback8
Router(config-subscriber)# commit
```

Running Configuration

An example of the running configuration for the policy-map with HTTP enrichment action is shown below:

```
policy-map type pbr http-enrichment-policy
 class type traffic open-garden-class
  transmit
 !
 class type traffic http-enrich-class1
  http-enrichment parameter-list subscriber-mac hostname bng-interface
 !

 class class-default
  drop
 !
!
end-policy-map
```

The running configuration for the subscriber enrichment parameter list is shown below:

```
subscriber
  http enrichment parmeter-list subscriber-mac subscriber-ip hostname bng-interface
!
```

The running configuration for the subscriber BNG identifier interface is shown below:

```
subscriber
 bng-interface Loopback8
!
```

Verification

The HTTP header enrichment statistics can be obtained using the commands **show policy-map type pbr interface all** and **show pbr internal statistics** .

```
Router# show policy-map type pbr interface all
node0_3_CPU0: (null): Service Policy not installed
node0_2_CPU0: (null): Service Policy not installed
node0_1_CPU0: (null): Service Policy not installed
node0_0_CPU0: (null): Service Policy not installed
node0_RSP0_CPU0:
Bundle-Ether20.1.ip296 input: httpr-policy-pldt
Policy Name: httpr-policy-pldt

Class httpr-class-pldt
  Classification statistics             (packets)
    Matched             :                  13
  Http enrichment statistics         (packets)
```

```
    HTTP Responses Sent   :                  13
    HTTP Request Sent      :                  6
   TCP packets Sent         :                  7
    Enrichment drops    :                     0
Class class-default
  Classification statistics          (packets/bytes) (May be 10secs old)
    Matched             :                308/31094

Router# show pbr internal statistics location 0/1/cpU0
PBR EA Internal Statistics:
  RSI Replay End Pending: 0
  Database:
    IFH Add:                        1
    IFH Delete:                     1
    IFH Total:                      0
    Mutex Block:                    0
  IM:
    Caps Add:                       1
    Caps Add Error:                 0
    Caps Remove:                    1
    Caps Remove Error:              0
    DPC:                            4
    DPC Error:                      0
    Init Data Update:               0
    Init Data Update Error:         0
  Switching:
    Rx HTTPR TCP SYN:               0
    Rx HTTPR TCP ACK:               0
    Rx HTTPR TCP FIN:               0
    Rx HTTPR HTTP GET:              0
    Rx HTTPR HTTP HEAD:             0
    Rx HTTPR HTTP POST:             0
    Tx HTTPR TCP SYN ACK:           0
    Tx HTTPR TCP FIN ACK:           0
    Tx HTTPR HTTP Redirect:         0
    Tx HTTP Enrichment TCP ACK:     8
    Tx HTTP Enrichment TCP FIN:     0
    Tx HTTP Enrichment TCP SYN:     2
    Tx HTTP Enrichment TCP RST:     3
    Tx HTTP Enrichment TCP SYN ACK: 0
    Tx HTTP Enrichment TCP FIN ACK: 0
    Tx HTTP Enrichment TCP RST ACK: 1
    Tx HTTP Enrichment Response:    0
    Tx HTTP Enrichment Request:    10
    DROP IFH Class ID Inval:        0
    DROP API Error:                 0
    DROP No Connection:             0
    DROP HTTPR Fragment:            0
    DROP HTTPR IP Options:          0
    DROP HTTPR TCP ACK:             0
    DROP HTTPR TCP Parse:           0
    DROP HTTPR HTTP Parse:          0
    DROP HTTP Enrichment Fragment:  0
    DROP HTTP Enrichment IP Option: 0
    DROP HTTP Enrichment TCP ACK:  14
    DROP HTTP Enrichment TCP Parse  0
    DROP HTTP Enrichment HTTP Parse:0
```

Idle Timeout for IPoE and PPPoE Sessions

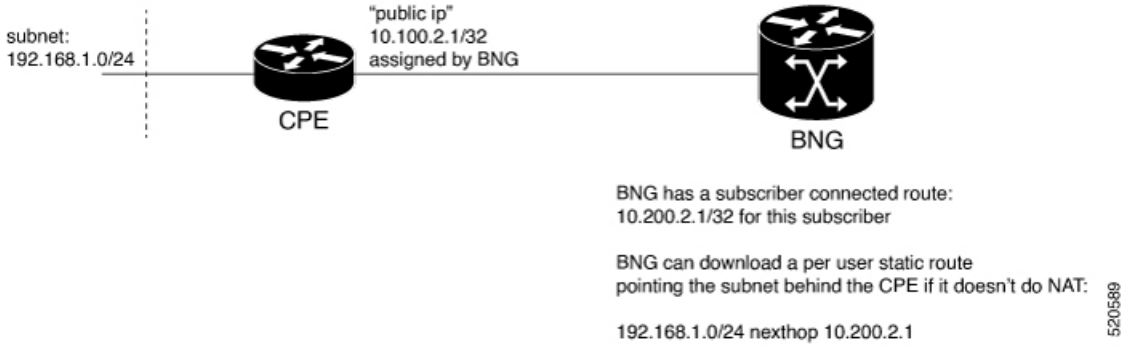
The Idle Timeout feature for IPoE and PPPoE sessions allows users to configure a maximum period of time that the subscriber sessions may remain idle. The subscriber sessions are terminated when this timeout period expires. The BNG monitors both the ingress and egress traffic for the determination of the idle time for the subscriber sessions. Control packets are not considered while determining session inactivity.

For more information about the Idle Timeout for IPoE and PPPoE Sessions feature, see the *Configuring Subscriber Features* chapter in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide*. For complete command reference for this feature, see the *Control Policy Commands* chapter and *Dynamic Template Commands* chapter in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*.

Routing Support on Subscriber Sessions

Routing support on subscriber sessions allows dynamic routes to be added on an individual subscriber basis for IPoE and PPPoE sessions. This allows forwarding traffic to a subscriber subnet that resides behind the customer premise equipment (CPE).

Here is a typical scenario:



Restrictions

- For IPv6 when a next hop and tag values are defined, you need to configure the metric value for the subscriber to be connected. If not defined, the subscriber fails to connect.
- For IPv6 when only the tag value is set and no metric is configured, the value is interpreted as a metric. The route fails to install if the tag is greater than 254, but the subscriber will be able to connect.

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.

For more information about the Traffic Mirroring on Subscriber Session feature, see the *Configuring Subscriber Features* chapter in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide*. For complete command reference for this feature, see the *Dynamic Template Commands* chapter in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*.

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 80

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. Use the **commit** or **end** command.
5. **test radius coa** {**activate** | **deactivate**} **service** *name* **acct-ses-id** *name*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
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`
		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.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-dynamic-template-type)# mirror first 100`	
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
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) subordinate to primary 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
```

Controlling Subscriber Plans Using Protocol Options

The BNG router supports identity change for DHCP Remote-Id parameter, thereby allowing modification of the subscriber service. This functionality of controlling subscriber plans using protocol options helps to deactivate an old service and activate a new service for the subscriber, based on the new service name received in Remote-Id parameter.

For more information about this feature, see the *Configuring Subscriber Features* chapter in the *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide*. For complete command reference, see the *Subscriber Commands* chapter in *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Command Reference*.

How to Control Subscriber Plans Using Protocol Options

To enable identity change handling across BNG router, use **subscriber featurette identity-change** command in the global configuration mode. A new event, **session-identity-change**, is introduced to specify an identity change event. In order to decode the service encoded in the DHCP Remote-Id parameter, as per the defined AAA attribute format, use **decode** command in the policy-map event class configuration mode. The **activate** command in the class-map sub-configuration mode is also enhanced to include an option **service-from-line** to activate the service encoded in the DHCP Remote-Id parameter.

Consider an example where the Remote-Id parameter received by BNG is in the format '*Username*|*Service-Name*', say, *bng_user*|*service1*. At session start stage, DHCP passes this Remote-Id in the 'parameters-to-add' list. The 'parameters-to-remove' list is empty at that point of time. The decode instruction decodes the Remote-Id based on the defined AAA attribute format (my-format in this example), extracts Username (bng_user) and Service-Name (service1) from it, and adds these parameters to the system. The **service-from-line** parameter refers to this particular Service-Name. The extracted Service-Name is added to AAA_AT_SUBSCRIBER_ACTIVATE_SERVICE attribute in the transaction context and it gets activated.

FSOL clients trigger an event when they want to change an identity parameter. DHCP triggers this event only if identity-change feature is enabled. In this event, DHCP passes the new Remote-Id (say, *bng_user*|*service2*) in parameters-to-add list and passes the old Remote-Id as part of parameters-to-remove list. The new Service-Name to be activated (service2) and the old Service-Name (service1) to be deactivated are decoded, extracted and added to AAA_AT_SUBSCRIBER_ACTIVATE_SERVICE and

AAA_AT_SUBSCRIBER_DEACTIVATE_SERVICE attributes respectively, in the transaction context. The old Service-Name gets deactivated and the new Service-Name gets activated.

Running Configuration

```
configure

 /* This configuration defines the format of Remote-Id from which BNG can derive the Username
 and Service-Name */
 aaa attribute format my-format format-string length 253 "%[^|]|%[^|]|" username service-name

 !

 /* Enable identity change feature */
 subscriber
  featurette identity-change
  !
 !

policy-map type control subscriber pm-dtmpl-1
 event session-start match-first
  class type control subscriber CL1 do-until-failure
   1 decode remote-id format my-format
   2 activate dynamic-template service-from-line
   3 authorize aaa list default identifier remote-id password blank

event session-identity-change match-first
  class type control subscriber CL1 do-until-failure
   1 decode remote-id format my-format
   2 deactivate dynamic-template service-from-line
   3 activate dynamic-template service-from-line
   4 authorize aaa list default identifier remote-id password blank
```

Whenever the **deactivate dynamic-template service-from-line** is used after the **decode** instruction, it has to be placed before **activate** or **authorize** commands.

Verification

Use these commands to verify the activation and deactivation of old and new services for the subscriber:

- **show subscriber session all detail**
- **show dhcp ipv6 server binding detail**
- **show policy-map applied interface bundle-ether** *interface-name*

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