

CGv6 Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 26.1.x, 26.2.x, 26.3.x

First Published: 2026-02-28

Last Modified: 2026-09-11

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE

Preface ix

Changes to This Document ix

Communications, Services, and Additional Information ix

CHAPTER 1

New and Changed Carrier Grade NAT Feature 1

Carrier Grade NAT Features Added or Modified in IOS XR Release 25.x.x 1

CHAPTER 2

Introduction 3

Overview and Benefits 3

Overview 3

Benefits of CGv6 3

IPv4 Address Shortage 3

Prerequisites for Implementing the CGv6 4

Implementation of NAT 4

Implementing NAT with ICMP 4

ICMP Query Session Timeout 4

Implementing NAT with TCP 5

Address and Port Mapping Behavior 5

Internally Initiated Connections 5

Externally Initiated Connections 5

Double NAT 444 5

Address Family Translation 5

Jumbo Frame Support 6

CHAPTER 3

Carrier Grade IPv6 over Virtualized Services Module (VSM) 7

Virtualized Services Module 7

Installing CGv6 on VSM 8
Prerequisites 9
Installing CGv6 OVA Package 9
Activating CGv6 VM 10
Uninstalling CGv6 on VSM 12
Upgrading CGv6 OVA Package 13
VSM scale numbers 15
Implementing NAT 64 over VSM 16
NAT44 and NAT 64 over VSM 19
TCP Sequence Check 20
Address and Port-Dependent Filtering 21
NAT0 or NAT Bypass Mode 21
Static Destination Address Translation 22
Multiple NetFlow and Syslog Servers for CGN Logging 23
Implementing NAT44 or NAT64 over VSM 23
Configuring NAT44 over VSM 23
Configuring a NAT44 Instance 23
Configuring the Application Service Virtual Interface (NAT44) 25
Configuring an Inside and Outside Address Pool Map (NAT44) 26
Predefined NAT 28
Considerations and Limitations of Predefined NAT 28
Configuring the Predefined Mode for NAT44 29
Configuring NAT64 over VSM 31
Configuring a NAT64 Instance 31
Configuring the Application Service Virtual Interface (NAT64) 32
Configuring an Inside and Outside Address Pool Map (NAT64) 34
Policy Functions 37
Configuring Port Limit per Subscriber 37
Configuring the Timeout Value for ICMP, TCP and UDP Sessions 38
FTP-ALG 39
RTSP-ALG 39
PPTP-ALG 40
TCP Maximum Segment Size Adjustment 40
Static Port Forwarding 40

Configuring Dynamic Port Range 41
(Only NAT44) Configuring One-to-One Mapping 42
(Only NAT44) Configuring TCP Sequence Check 43
(Only NAT44) Enabling Address and Port-Dependent Filtering 45
(Only NAT44) Configuring NAT0 Mode 46
(Only NAT44) Configuring the Static Destination Address Translation 48
(Only NAT44) Configuring Multiple NetFlow Servers 50
(Only NAT64) Configuring External Logging for NetFlow Servers 52
(Only NAT44) Configuring Multiple Syslog Servers 54
(Only NAT44) Configuring External Logging for the NAT Table Entries 57
Configuring the Server Address and Port for Netflow Logging 57
Configuring the Path Maximum Transmission Unit for Netflow Logging 58
Configuring the Refresh Rate for Netflow Logging 60
Configuring the Timeout for Netflow Logging 62
(Only NAT44) Syslog Logging 64
Configuring the Server Address and Port for Syslog Logging 64
Configuring the Host-Name for Syslog Logging 65
Configuring the Path Maximum Transmission Unit for Syslog Logging 67
Traffic Flow Mirroring 69
Limitations and Assumptions 73
Configuring Mirroring Using Destination Address Filter and Collector IP Address 73
Configuring Mirroring Using Destination Address, Port Number, Protocol Type, Source-Prefix Filters, and Collector IP Address 75
Configuring Mirroring for Dropped Packets Using Collector IP Address 77
Mapping of Address and Port-Encapsulation Mode 80
Configuring MAP-E 80
Configuring the Application Service Virtual Interface 80
Configuring a MAP-E Instance 82
Configuring the Policy Functions 83
Configuring Address Family 83
Configuring IPv4 Address Family 83
Configuring IPv6 Address Family 86
Configuring AFTR Endpoint Address 89
Configuring Contiguous Ports 90

Configuring CPE Domain Parameters 91
Configuring Path MTU of the Tunnel 93
Configuring Port Sharing Ratio 94
MAP-E on VSM Configuration: Example 95
Mapping of Address and Port-Translation Mode 96
Configuring MAP-T 97
Configuring the Application Service Virtual Interface 97
Configuring a MAP-T Instance 99
Configuring the Policy Functions 100
Configuring Address Family 100
MAP-T Configuration on VSM: Example 115
Configuration Examples for Implementing the Carrier Grade NAT 115
Configuring a Different Inside VRF Map to a Different Outside VRF: Example 116
Configuring a Different Inside VRF Map to a Same Outside VRF: Example 117
NAT44 Configuration: Example 117
NAT64 Configuration: Example 119
Bulk Port Allocation and Syslog Configuration: Example 121
Predefined NAT Configuration: Example 121
PPTP ALG Configuration: Example 122
NAT44 Instance 122
DBL Configuration: Example 122
NAT44 Instance 122
Configuring TCP Sequence-Check: Example 122
Configuring Address and Port-Dependent Filtering: Example 122
NAT0 Mode Configuration: Example 122
Configuration of Multiple NetFlow Servers: Example 123
Configuration of Multiple Syslog Servers: Servers 123
CGN Sequential Allocation Algorithm 124
Limitations 124
Configuring Sequential Predefined NAT 125

CHAPTER 4 **Carrier Grade IPv6 without Service Modules** 129
MAP-E without service modules 129
Configuring MAP-E without service modules 130

Configuring MAP-E instances without service modules 130
Configuring cpe domain parameters without service modules 132
Configuring port sharing ratio and contiguous port without service modules 133
Configuring BR Endpoint Address without modules 135
MAP-T without Service Cards 137
Configuring MAP-T without Service Cards 138
MAP-T Enhancements 142
Limitations and Scale Statistics 146
Configuring MAP-T Enhancements 147
TCP MSS for MAP-T 154
Configure TCP MSS for MAP-T 154
Scaling Up of MAP-T Instances 156
Configure TCAM for MAP-T Traffic 156
Overview of MAP-T Logging 158

CHAPTER 5 **Understanding MAP-T Exception Handling** 161
Configuration and Scalability Limits 161
Supported IPv6 Extension Headers 162
Processing of Fragmented Packets 162
Configuring MAP-T for Exception Handling with VSM 163
Running Configuration 163
Troubleshooting ICMP Errors 165
Translating ICMP IPv4 Headers to ICMP IPv6 Headers 165
Translating ICMP IPv6 Headers to ICMP IPv4 Headers 166

CHAPTER 6 **External Logging** 169
Bulk Port Allocation 169
Restrictions for Bulk Port Allocation 169
Session logging 170
(Only NAT44) Syslog Logging 170
Restrictions for Syslog 170
Syslog Message Format 170
Header 170
Structured Data 171

MSG 171
Netflow v9 Support 174
NetFlow Record Format 174
Reliable Log Transfer 187
Reliable Log Transfer Configuration: Examples 188
Frequently Asked Questions (FAQs) 189



Preface

The Preface contains these sections:

- Changes to This Document, on page ix
- Communications, Services, and Additional Information, on page ix

Changes to This Document

The following table lists the technical changes made to this document since it was first published.

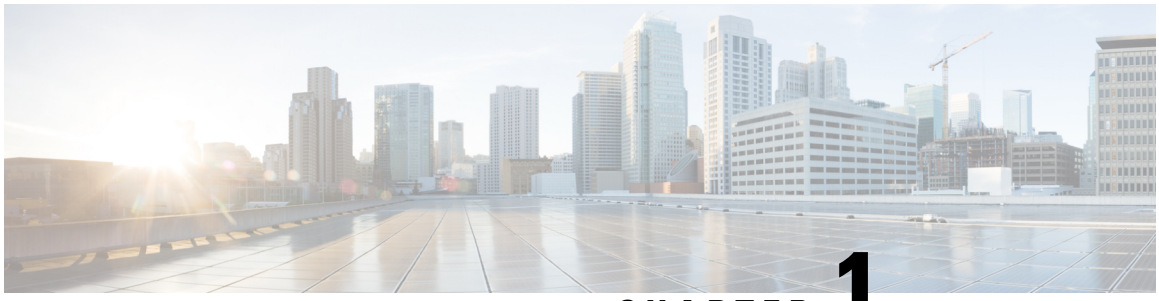
Date	Change Summary
February 2025	Initial release of this document

Communications, Services, and Additional Information

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

Cisco Bug Search Tool

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



CHAPTER 1

New and Changed Carrier Grade NAT Feature

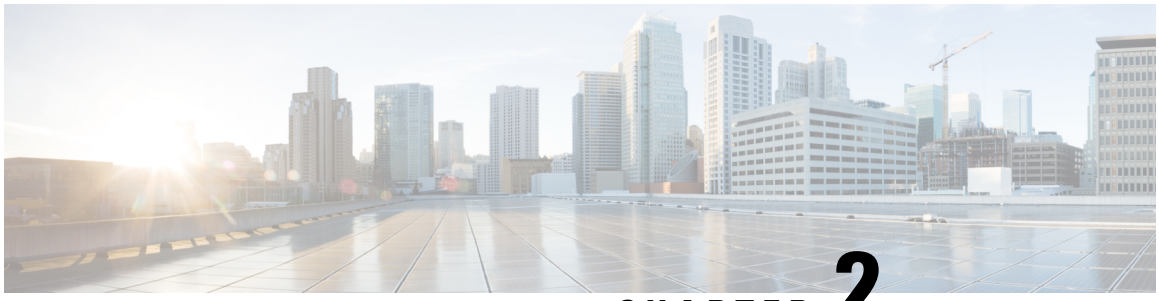
This table summarizes the new and changed information for the , and tells you where the features are documented.

- Carrier Grade NAT Features Added or Modified in IOS XR Release 25.x.x, on page 1

Carrier Grade NAT Features Added or Modified in IOS XR Release 25.x.x

Table 1: New and Changed Career Grade NAT Features

Feature	Description	Changed in Release	Where Documented
None	No new features introduced	Not applicable	Not applicable



CHAPTER 2

Introduction

The following table lists changes made to the document.

- Overview and Benefits, on page 3
- Prerequisites for Implementing the CGv6, on page 4
- Implementation of NAT, on page 4
- Double NAT 444, on page 5
- Address Family Translation, on page 5
- Jumbo Frame Support, on page 6

Overview and Benefits

Overview

Benefits of CGv6

CGv6 offers these benefits.

- Enables service providers to execute orderly transitions to IPv6 through mixed IPv4 and IPv6 networks.
- Provides address family translation but not limited to just translation within one address family.
- Delivers a comprehensive solution suite for IP address management and IPv6 transition.

IPv4 Address Shortage

A fixed-size resource such as the 32-bit public IPv4 address space will run out in a few years. Therefore, the IPv4 address shortage presents a significant and major challenge to all service providers who depend on large blocks of public or private IPv4 addresses for provisioning and managing their customers.

Service providers cannot easily allocate sufficient public IPv4 address space to support new customers that need to access the public IPv4 Internet.

Prerequisites for Implementing the CGv6

The following prerequisites are required to implement CGv6.

- You must be running Cisco IOS XR software Release 4.2.0 and above.
- You must have installed the CGv6 service package, **asr9k-services-p.pie** (to be used with RSP2) or **asr9k-services-px.pie** (to be used with RSP3).
- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command.

Note All the error conditions result in a syslog message. On observation of Heartbeat failure messages, contact Cisco Technical Support with **show tech-support services cgn** information.

Note If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Implementation of NAT

This section explains various implementations of NAT. The implementation of NAT over ISM and VSM are explained in the following chapters.

Implementing NAT with ICMP

This section explains how the Network Address Translation (NAT) devices work in conjunction with Internet Control Message Protocol (ICMP).

The implementations of NAT varies in terms of how they handle different traffic.

ICMP Query Session Timeout

RFC 5508 provides ICMP Query Session timeouts. A mapping timeout is maintained by NATs for ICMP queries that traverse them. The ICMP Query Session timeout is the period during which a mapping will stay active without packets traversing the NATs. The timeouts can be set as either Maximum Round Trip Time (Maximum RTT) or Maximum Segment Lifetime (MSL). For the purpose of constraining the maximum RTT, the Maximum Segment Lifetime (MSL) is considered a guideline to set packet lifetime.

If the ICMP NAT session timeout is set to a very large duration (240 seconds) it can tie up precious NAT resources such as Query mappings and NAT Sessions for the whole duration. Also, if the timeout is set to very low it can result in premature freeing of NAT resources and applications failing to complete gracefully. The ICMP Query session timeout needs to be a balance between the two extremes. A 60-second timeout is a balance between the two extremes.

Implementing NAT with TCP

This section explains the various NAT behaviors that are applicable to TCP connection initiation. The detailed NAT with TCP functionality is defined in RFC 5382.

Address and Port Mapping Behavior

A NAT translates packets for each TCP connection using the mapping. A mapping is dynamically allocated for connections initiated from the internal side, and potentially reused for certain connections later.

Internally Initiated Connections

A TCP connection is initiated by internal endpoints through a NAT by sending SYN packet. All the external IP address and port used for translation for that connection are defined in the mapping.

Generally for the client-server applications where an internal client initiates the connection to an external server, to translate the outbound SYN, the resulting inbound SYN-ACK response mapping is used, the subsequent outbound ACK, and other packets for the connection.

The 3-way handshake corresponds to method of connection initiation.

Externally Initiated Connections

For the first connection that is initiated by an internal endpoint NAT allocates the mapping. For some situations, the NAT policy may allow reusing of this mapping for connection initiated from the external side to the internal endpoint.

Double NAT 444

The Double NAT 444 solution offers the fastest and simplest way to address the IPv4 depletion problem without requiring an upgrade to IPv6 anywhere in the network. Service providers can continue offering new IPv4 customers access to the public IPv4 Internet by using private IPv4 address blocks, if the service provider is large enough; However, they need to have an overlapping RFC 1918 address space, which forces the service provider to partition their network management systems and creates complexity with access control lists (ACL).

Double NAT 444 uses the edge NAT and CGN to hold the translation state for each session. For example, both NATs must hold 100 entries in their respective translation tables if all the hosts in the residence of a subscriber have 100 connections to hosts on the Internet). There is no easy way for a private IPv4 host to communicate with the CGN to learn its public IP address and port information or to configure a static incoming port forwarding.

Address Family Translation

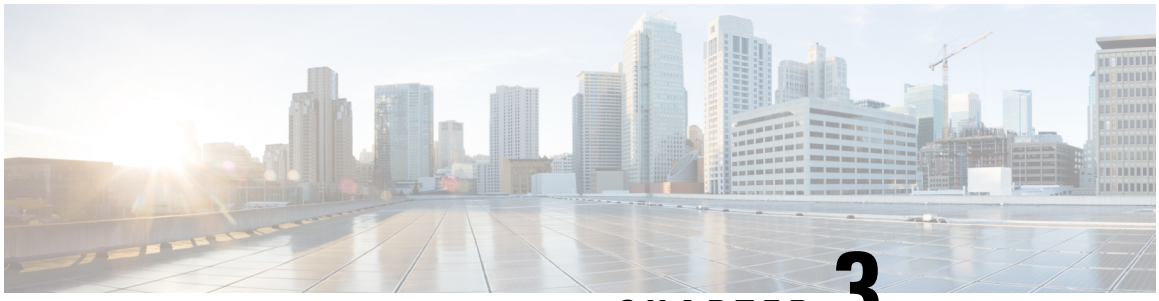
The IPv6-only to IPv4-only protocol is referred to as address family translation (AFT). The AFT translates the IP address from one address family into another address family. For example, IPv6 to IPv4 translation is called NAT 64 or IPv4 to IPv6 translation is called NAT 46.

Jumbo Frame Support

Jumbo frames are frames that are larger than the standard Ethernet frame size, which is 1518 bytes. The definition of frame size is vendor-dependent, and are not part of the IEEE standard.

The Integrated Services Module (ISM) and Virtualized Services Module (VSM) both support Jumbo Frames.

To enable Jumbo Frame support, configure the Maximum Transmission Unit (MTU) value of both the ingress and egress interfaces. The default MTU value is 1512 bytes and the maximum value is 9216 bytes.



CHAPTER 3

Carrier Grade IPv6 over Virtualized Services Module (VSM)

This module describes how to implement the Carrier Grade IPv6 (CGv6) over Virtualized Services Module (VSM).

- Virtualized Services Module, on page 7
- Installing CGv6 on VSM , on page 8
- Uninstalling CGv6 on VSM, on page 12
- Upgrading CGv6 OVA Package, on page 13
- VSM scale numbers, on page 15
- Implementing NAT 64 over VSM, on page 16
- NAT44 and NAT 64 over VSM, on page 19
- Implementing NAT44 or NAT64 over VSM, on page 23
- Traffic Flow Mirroring, on page 69
- Mapping of Address and Port-Encapsulation Mode, on page 80
- Configuring MAP-E, on page 80
- MAP-E on VSM Configuration: Example, on page 95
- Mapping of Address and Port-Translation Mode, on page 96
- Configuring MAP-T, on page 97
- MAP-T Configuration on VSM: Example, on page 115
- Configuration Examples for Implementing the Carrier Grade NAT, on page 115
- Configuring TCP Sequence-Check: Example, on page 122
- Configuring Address and Port-Dependent Filtering: Example, on page 122
- NAT0 Mode Configuration: Example, on page 122
- Configuration of Multiple NetFlow Servers: Example, on page 123
- Configuration of Multiple Syslog Servers: Servers, on page 123
- CGN Sequential Allocation Algorithm, on page 124

Virtualized Services Module

Virtualized Services Module (VSM) is the next generation service card on the Cisco ASR 9000 Series Aggregation Services Router. The software infrastructure on this card provides a virtual environment and the services run as virtual machines (VM) in this environment. The VMs simulate individual physical computing

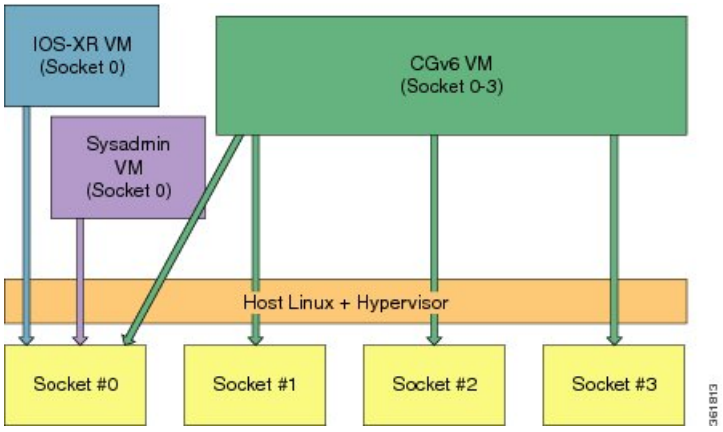
environments over a common hardware. The available hardware resources, like processor, memory, hard disk, and so on, are virtualized and allocated to individual virtual machines by the hypervisor.

Note A9K-VSM-500 is not supported on Cisco IOS XR 64-bit image.

VSM Components

VSM is capable of hosting multiple VMs. It consists of the following components:

- IOS XR VM: This VM is used for managing the routing functions.
- System Admin VM: This VM is used for the system administration
- Application VM: CGv6 is the application VM running on VSM. In the current release, only one CGv6 VM can run at a given time.
- Linux Host and Hypervisor: The routing functions and the system administration functions are run on separate virtual machines (VMs) over a Linux host operating system. The CGv6 VM, along with the other VMs, runs on the top of the KVM hypervisor.



Features and Considerations

Some of the features and considerations of VSM are:

- The CGv6 application has to run in a VM environment.
- The IOS XR Service Enablement CLIs are needed to create, delete, access, and operate on CGv6 VM.
- The VSM card can co-exist with other LCs including ISM.
- Each NP has 6 NP ports and can send traffic to 24 CGv6 Application processes.
- For each VSM card, a ServiceInfra interface needs to be configured.
- Traffic diversion may be done based on a static route or ACL-based forwarding (ABF).
- In the current release, VSM does not support multiple CGv6 VMs on the same card.

Installing CGv6 on VSM

The process of installing CGv6 on VSM involves the following:

Note If you are performing an upgrade or a downgrade, the CGv6 VM needs to be deactivated first, uninstalled, installed, and then activated. See Uninstalling CGv6 on VSM for information on deactivating the CGv6 VM and uninstalling the CGv6 OVA package.

Prerequisites

Ensure that you have installed the following images:

- asr9k-mini-px.vm (Base IOS-XR image)
- asr9k-services-infra.pie (VSM Services Infra package)
- asr9k-services-px.pie (CGv6 Services package)
- asr9k-fpd-px.pie (FPGA Image IOS XR package)
- asr9k-vsm-cgv6-<version>.ova (Linux Open Virtual Alliance or OVA package)

Installing CGv6 OVA Package

The CGv6 Virtual Machine (VM) is provided as an OVA package. Open Virtualization Appliance (OVA) is a single file distribution of the file package. The CGv6 OVA package consists of the following files:

- OVA Profile Descriptor file
- Package version file
- Linux Image file

The process of installation of CGv6 OVA package consists of the following steps:

1. Copy the OVA file from the remote location to the RP disk.

Note Once the CGv6 OVA package is copied to RP's disk, you can install it on multiple VSMs on the same chasis.

2. Install CGv6 VM on a specific VSM card.

```
RP/0/RSP0/CPU0:router# virtual-service install name <service/VM name> package <OVA package
 name> node <VSM_location>
```

Note The service or VM name can contain only alphanumeric characters (A to Z, a to z, or 0 to 9), or an underscore (_). All other special characters are not allowed. The installation process might take about 7-8 minutes.

3. Check the progress of the installation process by using the show virtual-service list command. Once the installation is complete, the status is changed to Installed.

```
RP/0/RSP0/CPU0:router# show virtual-service list
```

```
Virtual Service List:
         Name     Status        Package Name          Node
         cgn1     Installing    asr9k-vsm-cgv6.ova    0/1/CPU0

RP/0/RSP0/CPU0:NAT#sh virtual-service list

Virtual Service List:
         Name     Status        Package Name          Node
         cgn1     Installed    asr9k-vsm-cgv6.ova    0/1/CPU0
```

Activating CGv6 VM

The steps to activate the CGv6 VM are as follows:

1. Configure the CGv6 VM and the 12 Gigabit Ethernet (GE) interfaces in the global configuration mode.

```
RP/0/RSP0/CPU0:router(config)# virtual-service cgn123

RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/2/1/0

RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/2/1/1

RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/2/1/2

RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/2/1/3

RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/2/1/4

RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/2/1/5

RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/2/1/6

RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/2/1/7

RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/2/1/8

RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/2/1/9

RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/2/1/10

RP/0/RSP0/CPU0:router(config-virt-service)# vnic interface tenGigE 0/2/1/11

RP/0/RSP0/CPU0:router(config-virt-service)# commit
```

2. Activate the CGv6 VM.

```
RP/0/RSP0/CPU0:router(config-virt-service)# activate
RP/0/RSP0/CPU0:router(config-virt-service)#commit
```

3. Check the progress of the activation process by using the show virtual-service list command. Once the VM is activated, the status changes to Activated.

```
RP/0/RSP0/CPU0:router# show virtual-service list

Virtual Service List:
         Name     Status        Package Name
         cgn1     Activated    asr9k-vsm-cgv6.ova
```

Note Once the VM is activated, it takes about 5 minutes for the CGv6 applications to come up.

4. Configure the ServiceInfra interface.

```
RP/0/RSP0/CPU0:router# configure terminal

RP/0/RSP0/CPU0:router(config)# interface ServiceInfra 1
```

```
RP/0/RSP0/CPU0:router(config-int)# ipv4 address 3.1.1.1 255.255.255.252
RP/0/RSP0/CPU0:router(config-int)# service-location 0/2/CPU0
RP/0/RSP0/CPU0:router(config-int)# commit
```

5. Before you configure NAT44, ensure that the 12 Gigabit Ethernet (GE) interfaces are up. If they are in the shutdown mode, then change their mode by using the **no shut** command.

Note In IOS-XR, by default, any interface that is not configured is shut down when the associated line card is reloaded. To prevent this behavior on the VSM TenGigE interface (port), add a minor configuration (such as, *description*) on the interface.

```
RP/0/RSP0/CPU0:router(config)# interface tenGigE 0/2/1/0
RP/0/RSP0/CPU0:router(config-if)# interface tenGigE 0/2/1/1
RP/0/RSP0/CPU0:router(config-if)# interface tenGigE 0/2/1/2
RP/0/RSP0/CPU0:router(config-if)# interface tenGigE 0/2/1/3
RP/0/RSP0/CPU0:router(config-if)# interface tenGigE 0/2/1/4
RP/0/RSP0/CPU0:router(config-if)# interface tenGigE 0/2/1/5
RP/0/RSP0/CPU0:router(config-if)# interface tenGigE 0/2/1/6
RP/0/RSP0/CPU0:router(config-if)# interface tenGigE 0/2/1/7
RP/0/RSP0/CPU0:router(config-if)# interface tenGigE 0/2/1/8
RP/0/RSP0/CPU0:router(config-if)# interface tenGigE 0/2/1/9
RP/0/RSP0/CPU0:router(config-if)# interface tenGigE 0/2/1/10
RP/0/RSP0/CPU0:router(config-if)# interface tenGigE 0/2/1/11
RP/0/RSP0/CPU0:router(config-if)# no shut
RP/0/RSP0/CPU0:router(config-if)# commit
```

Note When a virtual service is in *Activating* state and being recovered from a failure event like VSM card reload or server disconnect, and if another failure happens like the VSM card reload or server disconnect, then the virtual-service activation will not be attempted. In such conditions, virtual service is moved into *Activation Failed* state and needs to be recovered manually with the following sequence of operations:

```
RP/0/RSP0/CPU0:router# configure terminal
RP/0/RSP0/CPU0:router(config)# virtual-service cgn123
RP/0/RSP0/CPU0:router(config-virt-service)# no activate
RP/0/RSP0/CPU0:router(config-virt-service)# commit
RP/0/RSP0/CPU0:router(config-virt-service)# activate
RP/0/RSP0/CPU0:router(config-virt-service)# commit
```

Note After insertion of VSM card in chassis, you can install using the following CLI.
This command will install and automatically activate the CGN NAT44 services.

```
virtual-service autoActivate name <service_name> package <ova_location> location <VSM_location>
```

Alternatively, you can deactivate and uninstall the service by using the following CLI:

```
virtual-service autounistall name <service_name>
```

Ensure that you reload the VSM line card after uninstalling the VSM line card.

Uninstalling CGv6 on VSM

The process of uninstalling CGv6 VSM involves the following processes:

- Deactivating CGv6 VM
- Uninstalling CGv6 OVA Package
- Disabling the Service Enablement Feature

Deactivating CGv6 VM

To de-activate the CGv6 VM, perform the following in the global configuration mode:

```
RP/0/RP0/CPU0:router(config)# virtual-service cgn123
RP/0/RP0/CPU0:router(config-virt-service)# no activate
RP/0/RP0/CPU0:router(config-virt-service)# commit
```

After you have deactivated the CGv6 VM, reload the VSM card.

To remove the CGv6 instance, perform the following in the global configuration mode:

```
RP/0/RP0/CPU0:router(config)# no virtual-service cgn123
RP/0/RP0/CPU0:router(config)# commit
```

Uninstalling CGv6 OVA Package

To uninstall the CGv6 OVA package, run the following commands in the EXEC mode:

```
RP/0/RSP0/CPU0:router# virtual-service uninstall name cgn123
```

After you have uninstalled the CGv6 OVA package, reload the VSM card before installing any other OVA package.

Disabling the Service Enablement Feature

Note Perform this operation only if you are removing the CGN service completely from the router. If there are multiple VSM line cards in the chassis and you are removing or replacing an individual VSM card, do not perform this operation.

To disable the service enablement feature, run the following commands in the global configuration mode.

```
RP/0/RP0/CPU0:router(config)# no virtual-service enable
RP/0/RP0/CPU0:router(config)# commit
```

Upgrading CGv6 OVA Package

The CGv6 Virtual Machine (VM) is provided as an OVA package. Open Virtualization Appliance (OVA) is a single file distribution of the file package. The CGv6 OVA package consists of the following files:

- OVA Profile Descriptor file
- Package version file
- Linux Image file

The process of upgrading the CGv6 OVA package consists of the following steps:

1. Saving the virtual service configuration
2. Deactivating CGv6 VM
3. Uninstalling CGv6 OVA Package
4. Upgrading Cisco IOS XR
5. Installing CGv6 on VSM
6. Restoring the virtual service configuration

Note Before upgrading or downgrading the CGv6 OVA package on the Active VSM card in HA (high availability) mode, perform a graceful shift of the traffic from Active VSM to Standby VSM. This will ensure that the CGN-related configuration is replicated into a standby card. To perform graceful shift of the traffic, run the `service redundancy failover service-type all preferred-active` <*active-VSM-slot*> command in EXEC mode.

Saving the Virtual Service Configuration

Before you start upgrading the CGv6 OVA package, save the virtual service configuration and related data.

To see the virtual service configuration, perform the following in the global configuration mode:

```
RP/0/RP0/CPU0:router(config)# show run virtual-service
```

Deactivating CGv6 VM

To de-activate the CGv6 VM, perform the following in the global configuration mode:

```
RP/0/RP0/CPU0:router(config)#virtual-service cgn123
RP/0/RP0/CPU0:router(config-virt-service)#no activate
RP/0/RP0/CPU0:router(config-virt-service)#commit
```

After you have deactivated the CGv6 VM, reload the VSM card.

To remove the CGv6 instance, perform the following in the global configuration mode:

```
RP/0/RP0/CPU0:router(config)#no virtual-service cgn123
RP/0/RP0/CPU0:router(config)#commit
```

Uninstalling CGv6 OVA Package

To uninstall the CGv6 OVA package, run the following commands in the EXEC mode:

```
RP/0/RSP0/CPU0:router# virtual-service uninstall name cgn123
```

After you have uninstalled the CGv6 OVA package, reload the VSM card before installing any other OVA package.

Upgrading Cisco IOS XR

Upgrade Cisco IOS XR. For details refer the *Upgrading and Managing Cisco IOS XR software* chapter in

Installing the CGv6 OVA Package

The process of installing the CGv6 OVA package consists of the following steps:

1. Copy the OVA file from the remote location to the RP disk.

Note Once the CGv6 OVA package is copied to RP's disk, you can install it on multiple VSMs on the same chasis.

2. Install CGv6 VM on a specific VSM card.

```
RP/0/RSP0/CPU0:router#virtual-service install name <service/VM name> package <OVA package
 name> node <VSM_location>
```

Note The installation process might take about 7-8 minutes.

3. Check the progress of the installation process by using the show virtual-service list command. Once the installation is complete, the status is changed to Installed.

```
RP/0/RSP0/CPU0:router#sh virtual-service list

Virtual Service List:
         Name     Status        Package Name          Node
         cgn1     Installing    asr9k-vsm-cgv6.ova    0/1/CPU0

RP/0/RSP0/CPU0:NAT#sh virtual-service list

Virtual Service List:
         Name     Status        Package Name          Node
         cgn1     Installed    asr9k-vsm-cgv6.ova    0/1/CPU0
```

Restoring the Virtual Service Configuration

Restore the virtual service configuration.

CGv6 VM is activated automatically after the virtual service configuration is restored.

Check the progress of the activation process by using the show virtual-service list command. Once the VM is activated, the status changes to Activated.

```
RP/0/RSP0/CPU0:router# show virtual-service list

Virtual Service List:
         Name     Status        Package Name
         cgn1     Activated    asr9k-vsm-cgv6.ova
```

Note Once the VM is activated, it takes about 5 minutes for the CGv6 applications to come up.

VSM scale numbers

ASR9K supports the following VSM scale numbers:

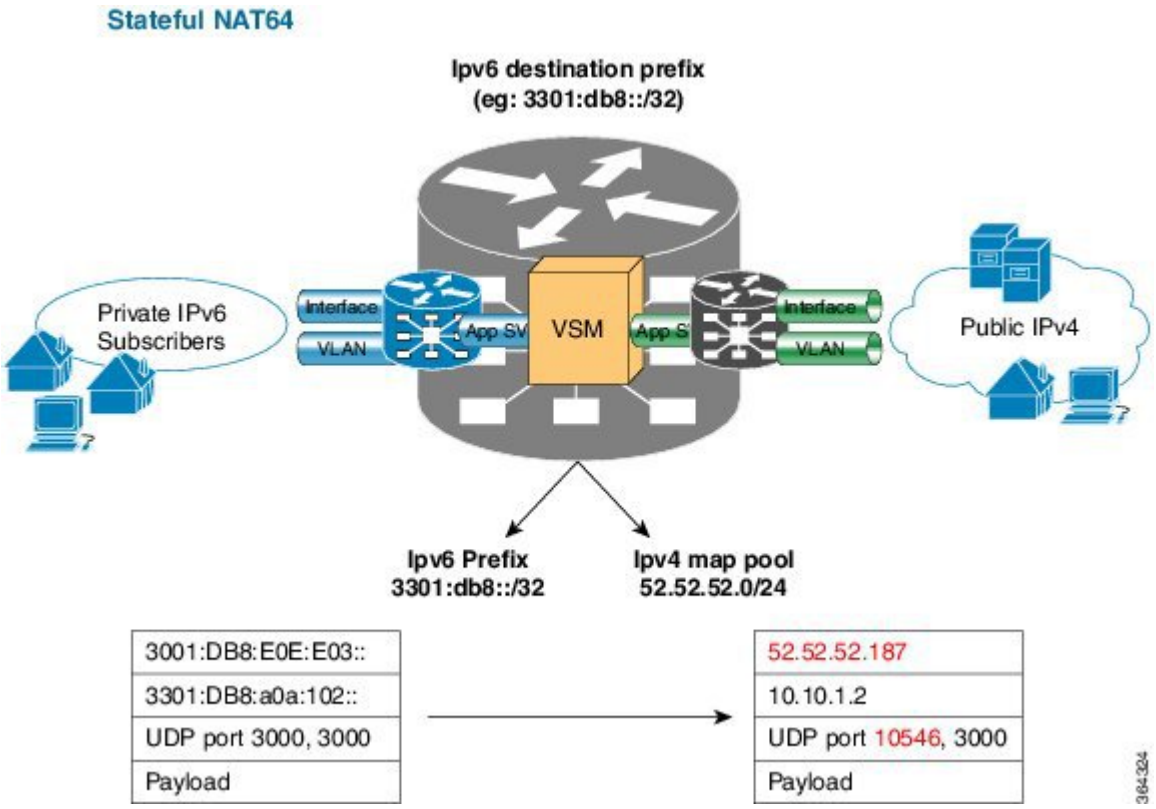
Parameter Name	Value per VSM	Value per ASR9K Chassis with VSM
Number of CGN or CGv6 Instances	1	8
Number of Service Infra Interfaces	1	8
Number of Service App interfaces	512	
Number of NAT44 instances	1	8
Number of Stateful Translation	80 Millions	
Number of NAT session	80 Million	
Number of NAT users	4 Million	
Number of Static Port Forwarding Entries	6000	
Number of Public IPv4 addresses	65536 or 16	
Number of VRF per NAT44 instance	128 (inside) + 128 (outside)	
BNG	32k per np	
VRF	8000	
DS-Lite Sessions	80 Million	
NAT64 Sessions	80 Million	

Parameter Name	Value per VSM	Value per ASR9K Chassis with VSM
6RD (ASR 9000 Enhanced Ethernet Line Card is inline with 6RD with an expectation rate of 90 percent.)		

Note Number of VSM cards per chassis can be adjusted based on the type of chassis and traffic assessment.

Implementing NAT 64 over VSM

This section explains how NAT64 is implemented over VSM. The figure illustrates the implementation of NAT64 over VSM.



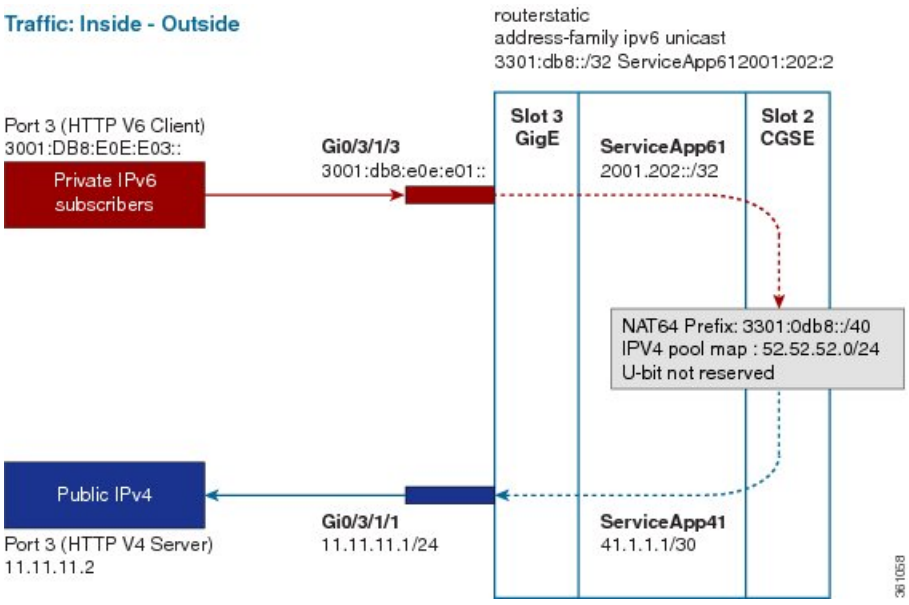
The components of this implementation are as follows:

- Private IP6 subscribers – It denotes a private network.
- Interface/VLAN- It denotes a designated interface or VLAN which is associated with the VRF.

- Inside VRF – It denotes the VRF that handles packets coming from the subscriber network. It is known as inside VRF as it forwards packets from the private network.
- App SVI- It denotes an application interface that forwards the data packet to and from the VSM. The data packet may be sent from another line card through a backplane. Because the VSM card does not have a physical interface, the APP SVI acts as a logical entry into it.

  The inside VRF is bound to an App SVI. There are 2 App SVIs required; one for the inside VRF and the other one for the outside VRF. Each App SVI pair will be associated with a unique "inside VRF" and a unique public IP address pool. The VRF consists of a static route for forwarding packets to App SVI1.
- Outside VRF- It denotes the VRF that handles packets going out to the public network. It is known as outside VRF as it forwards packets from the public network.
- Public IPV4- It denotes a public network.

The following figure illustrates the path of the data packet from a private network to a public network in a NAT64 implementation.

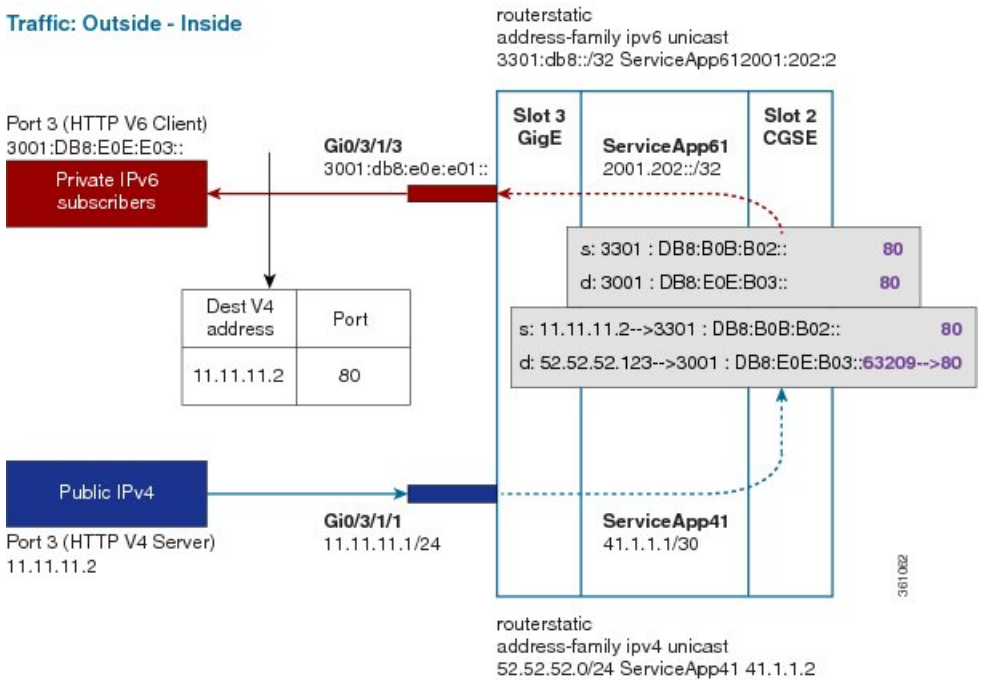


The packet goes through the following steps when it travels from the private network to the public network:

1. In the network shown in this figure, the packet travels from the host A (having the IP address 3001:DB8:E0E:E03::/40) in the private network to host B (having the IP address 11.11.11.2) in the public network. The private address has to be mapped to the public address by NAT64 that is implemented in VSM.
2. The packet enters through the ingress port on the Gigabit Ethernet (GigE) interface at Slot 3.
3. Once the packet reaches the designated interface or VLAN on ASR9K, it is forwarded to the inside VRF either through static routing or ACL-based forwarding (ABL). Based on this routing decision, the packet that needs address translation is determined and is forwarded to the App SVI that is bound to the VRF.
4. The packet is forwarded by AppSVI1 through a default static route. The destination address and the port get translated because of the CGN configuration applied on VSM.

5. The VSM applies NAT64 to the packet and a translation entry is created. The CGN determines the destination address from the FIB Look Up. It pushes the packet to the egress port.
6. The packet is then forwarded to the egress port on the interface through App SVI2. The packet is forwarded by App SVI2 through the default static route. Then the packet is sent to the public network.
7. The packets that do not need the address translation can bypass the App SVI and can be forwarded to the destination through a different static route and a different egress port.

The following figure illustrates the path of the packet coming from the public network to the private network.



The packet goes through the following steps when it travels from the public network to the private network:

1. In the network shown in this figure, the packet travels from the host A (having the IP address 11.11.11.2) in the public network to host B (having the IP address 3001:DB8:E0E:E03::) in the private network. The public address has to be mapped to the private address by NAT64 that is implemented in VSM.
2. The packet enters through the ingress port on the Gigabit Ethernet (GigE) interface at Slot 3.
3. Once the packet reaches the designated interface or VLAN on ASR9K, it is forwarded to the outside VRF either through static routing or ACL-based forwarding (ABL). Based on this routing decision, the packet is forwarded to the App SVI that is bound to the VRF.
4. The packet is forwarded by App SVI2 through a default static route. The destination address and the port are mapped to the translated address.
5. The VSM applies NAT64 to the packet. The CGN determines the destination address from the FIB Look Up. It pushes the packet to the egress port.
6. The packet is then forwarded to the egress port on the interface through App SVI2. Then the packet is sent to the private network through the inside VRF.

7. The packets that do not need the address translation can bypass the App SVI and can be forwarded to the destination through a different static route and a different egress port.

NAT44 and NAT 64 over VSM

The following are the features that are supported by NAT44 over VSM:

- TCP Sequence Check
- Address and Port-Dependent Filtering
- NAT0 or NAT Bypass Mode
- Static Destination Address Translation
- Multiple NetFlow and Syslog Servers for CGN Logging

The following are the features that are supported by NAT64 over VSM:

- Address and Port-Dependent Filtering
- Multiple NetFlow Servers for CGN Logging

VSM scale numbers supported in NAT 44

NAT 44 supports the following VSM scale numbers:

Parameter Name	Value per VSM	Value per ASR9K Chassis with VSM
Number of CGN or CGv6 Instances	1	8
Number of Service Infra Interfaces	1	8
Number of Service App interfaces	512	
Number of NAT44 instances	1	8
Number of Stateful Translation	80 Millions	80 x Number of Installed VSM cards based on type of chassis. Max 8 number of VSM cards supported per chassis.
Number of NAT session	80 Million	80 x Number of Installed VSM cards based on type of chassis. Max 8 number of VSM cards supported per chassis.

Parameter Name	Value per VSM	Value per ASR9K Chassis with VSM
Number of NAT users	4 Million	4 x Number of Installed VSM cards based on type of chassis. Max 8 number of VSM cards supported per chassis.
Number of Static Port Forwarding Entries	6000	6000 x Number of Installed VSM cards based on type of chassis. Max 8 number of VSM cards supported per chassis.
Number of Public IPv4 addresses	65536 or 16	
Number of VRF per NAT44 instance	128 (inside) + 128 (outside)	

VSM Scale numbers supported in NAT 64

NAT 64 supports the following VSM scale number:

Parameter Name	Value per VSM	Value per ASR9K Chassis with VSM
NAT64 Sessions	80 Millions	

TCP Sequence Check

In order to overcome security threats to less secure networks, Cisco Virtualized Services Module (VSM) performs TCP sequence check.

A sequence number is a 32-bit number that is included in a packet in a TCP session. The sequence numbers of the incoming packets are stored in the translation or session entry. If a packet's sequence number does not match the expected sequence number, then the packet is dropped . In this way, the networks can be secured from spoofed packets.

You can perform these TCP sequence checks by using the sequence-check command. An optional keyword, diff-window, has been provided for a user to define and configure the accepted expected range of sequence numbers. But it is recommended that the user does not specify this range and instead allows the router to compute the range for each TCP session based on the client-server negotiation.

Two counters are configured for the TCP sequence checks:

- Out-to-In packets counter: This counter keeps a count of the packets whose sequence numbers did not match the expected range. But yet these packets are translated and forwarded because TCP sequence check has not been configured.
- Dropped packets counter: This counter keeps a count of the packets that were dropped because of the TCP sequence check.

The counters are displayed by using the show cgn nat44 counters command.

Address and Port-Dependent Filtering

Currently, CGN on VSM implements the following by default:

- Endpoint-Independent Mapping: This mapping process reuses the port mapping for subsequent packets that are sent from the same internal IP address and port to any external IP address and port.
- Endpoint-Independent Filtering: This filtering process filters out only packets that are not destined to the internal address and port regardless of the external IP address and port source.

In such a configuration, by knowing the translated IP address and the port of a private host, any malicious host in a public network can initiate packet floods to that private host. In order to prevent such attacks, the address and port-dependent filtering feature has to be enabled by using the **filter-policy** command. The user can disable the filtering based on port by using the **ignore-port** keyword with this command.

Two counters are configured for the address and port-dependent filtering:

- Total number of sessions created due to Out2In packets: This counter keeps a count of the sessions that were created by the packets coming from outside.
- Number of Out2In drops due to end point filtering: This counter keeps a count of the packets that were dropped because of the endpoint filtering.

The counters are displayed by using the **show cgn nat44 counters** command.

NAT0 or NAT Bypass Mode

For some subscribers, a service provider may want to provide public addresses directly. Hence the Network Address Translation (NAT) is not required for these IP addresses. But at the same time, services like endpoint dependent filtering and TCP sequence check are required so that the subscribers do not receive any unwanted traffic from the Internet. In such cases, NAT0 or NAT Bypass mode is supported per inside-VRF.

The NAT0 mode along with the TCP sequence check and endpoint dependent filtering offers protection to the subscribers from Outside-to-Inside DoS attacks.

In this mode, when a subscriber sends a packet, an entry is made in the database. When the packet comes from the Outside-to-Inside direction, the entry is checked. But no translation occurs. The packet is allowed only if the entry exists.

NAT0 mode can be enabled by using the **nat-mode no-nat** command.

Considerations

Some of the considerations regarding the NAT0 mode are as follows:

- Static port forwarding is applicable in this mode.
- Bulk port allocation is not applicable in this mode.
- There is no need for the public address pool to be configured as the translation does not happen.
- If you have not initiated any traffic, then you will not receive any packet. Hence bandwidth can be saved and controlled by the public IP user.

Static Destination Address Translation

In static destination address translation, the VSM translates the destination IP address along with the source IP address. When a packet goes through NAT44, the source IP address translation happens. If the static destination address translation is enabled, then the destination IP address translation also takes place. For the source address translation, the IP address is taken from the public address pool. For the destination address translation, the user needs to provide a 1:1 mapping of the addresses in a .csv file.

To configure static destination address translation, use the **static-mapping -file direction** command. To delete the existing configuration use the **no static-mapping-file direction** command. If you want to modify the existing configuration, you can edit the existing .csv file and upload it back to the disk.

About the Static Mapping Configuration File

As mentioned before, the static mapping configuration file is in .csv format. The 1:1 mapping of the private addresses and the public addresses for the static destination address translation is provided in the .csv file. If this file becomes corrupt, then the destination address will not get translated. This file must be stored on the local disk.

The 1:1 mapping of the addresses must be in the following format in the .csv file:

```
13.1.1.0, 12.1.1.0
13.1.1.1, 12.1.1.1
13.1.1.2, 12.1.1.2
13.1.1.3, 12.1.1.3
13.1.1.4, 12.1.1.4
13.1.1.5, 12.1.1.5
```



Note The IP address, 13.1.1.0, is known as the premap IP address and the IP address, 12.1.1.0, as the postmap IP address

Considerations

Some of the considerations regarding the static destination address translation are as follows:

- The static destination address translation feature currently supports static destination NAT in the Inside-to-Outside direction and static source NAT in the Outside-to-Inside direction.
- Currently, the static destination address translation supports NAT44 only.
- The maximum value of entries for static mapping of addresses per VSM is 50K.

High Availability Support for the Route Switch Processor (RSP) Switchover

To ensure high availability during the RSP switchover, the user should manually copy the static mapping configuration file on both active RSP as well as the standby RSP.

VSM High Availability Support

The high availability of VSM is supported by default.

Multiple NetFlow and Syslog Servers for CGN Logging

Cisco IOS XR supports High Availability for NAT44 through multiple NetFlow or Syslog servers. Starting from Release 5.2.0, you can configure up to 8 additional NetFlow or Syslog Servers for a single inside VRF instance.

Note In an inside VRF with multiple servers configured, the NetFlow logging policy will be enabled on all servers if it is enabled for one or more servers inside the VRF.

Note In an inside VRF with multiple servers configured, the Syslog logging policy will be enabled on all servers if it is enabled for one or more servers inside the VRF.

Implementing NAT44 or NAT64 over VSM

This section explains the implementation of NAT44 or NAT64 on VSM.

Configuring NAT44 over VSM

Perform the following tasks to configure NAT44 on VSM:

1. Configuring a NAT44 Instance, on page 23
2. Configuring the Application Service Virtual Interface (NAT44), on page 25
3. #unique_43

Configuring a NAT44 Instance

Perform this task to configure a NAT44 instance.

Note The system does not support deleting VRF on live traffic in the following scenarios:

- If you are in the global configuration mode.
- If you are within the CGN instance.
- If you are in the static route table.

SUMMARY STEPS

1. **configure**
2. **service cgn nat44***instance-name*
3. **service-location preferred-active** *VSM location*

4. **service-type nat44 nat1**
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn nat44***instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 NAT44 application and enters CGv6 configuration mode.

Step 3	**service-location preferred-active** *VSM location* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-location preferred-active 0/3/CPU0`	Configures the NAT preferred active VSM location.
Step 4	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGv6 NAT44 application.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Application Service Virtual Interface (NAT44)

Perform this task to configure the application service virtual interface (SVI) to forward data traffic.

SUMMARY STEPS

1. **configure**
2. **interface ServiceApp** *value*
3. **service cgn** *instance-name* **service-type** *nat44*
4. **vrf** *vrf-name*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**interface ServiceApp** *value*
	Example:
	`RP/0/RP0/CPU0:router(config)# interface ServiceApp 1`
	`RP/0/RP0/CPU0:router(config-if)#`
		Configures the application SVI as 1 and enters interface configuration mode.

Step 3	**service cgn** *instance-name* **service-type** *nat44* **Example:** `RP/0/RP0/CPU0:router(config-if)# service cgn cgn1`	Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.
Step 4	**vrf** *vrf-name* **Example:** `RP/0/RP0/CPU0:router(config-if)# vrf insidevrf1`	Configures the VPN routing and forwarding (VRF) for the Service Application interface

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring an Inside and Outside Address Pool Map (NAT44)

Perform this task to configure an inside and outside address pool map with the following scenarios:



Note Do not configure multiple outside address-pools to be mapped to a single inside-vrf. If you have multiple outside address-pools to be mapped, then create multiple inside-vrfs and map each outside address-pool to a single inside-vrf inside the NAT44 configuration.

- The designated address pool is used for CNAT.
- One inside VRF is mapped to only one outside VRF.
- Multiple non-overlapping address pools can be used in a specified outside VRF mapped to different inside VRF.
- Max Outside public pool per CGSE/CGN instance is 64 K or 65536 addresses. That is, if a /16 address pool is mapped, then we cannot map any other pool to that particular CGSE.
- Multiple inside vrf cannot be mapped to same outside address pool.
- While Mapping Outside Pool minimum value for prefix is 16 and maximum value is 26.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **map** [**outside-vrf** *outside-vrf-name*] **address-pool** *address/prefix*
6. **end** or **commit**

DETAILED STEPS

Procedure

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

Step 2	**service cgn** *instance-name*
	Example:
	`Router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `Router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	Example:
	`Router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures an inside VRF named insidevrf1 and enters CGN inside VRF configuration mode.

Step 5	**map** [**outside-vrf** *outside-vrf-name*] **address-pool** *address/prefix*
	Example:
	`Router(config-cgn-invrf)# map outside-vrf outside vrf1 address-pool 10.10.0.0/16`
	`or`
	`Router(config-cgn-invrf)# map address-pool 100.1.0.0/16`
		Configures an inside VRF to an outside VRF and address pool mapping.

Step 6	**end** or **commit**
	Example:
	`Router(config-cgn-invrf-afi)# end`
	`or`
	`Router(config-cgn-invrf-afi)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Predefined NAT

In classic NAT, the process of mapping a private IP to a public IP or a private port to an outside port is random. Therefore, it becomes difficult to track the subscribers using an IP and a port at a given time. Predefined NAT avoids this random process by mapping a private IP address to a range of ports associated with the corresponding public IP address. This is done through an algorithm that helps the user to recognize a private IP address without having to refer to the massive CGN logs. The address and port translation is done in accordance with the algorithm.

In a predefined NAT configuration, if you want to trace a subscriber's private IP address from a public IP address and the associated port, perform the following steps:

- Whenever NAT is configured on a router or when there is a change in the existing configuration, use the following command to get the complete mapping information of private to public users:

  show cgn nat44 *instance-name* **mapping** {**inside-address | outside-address**} **inside-vrf***vrf-instance* **start-addr** *start address* [**end-addr***end address*]

  In the above command, specify the lowest address of the configured public IP pool as start address and the highest address of the pool as end address. This command dumps all the mapping for each private IP, the translated public IP, and port range. It is recommended that you divert this output in to a file and save it for future reference. Save this output to separate files each time you change the NAT44 configuration parameters and note down the time at which the changes were made and the corresponding file name.

- Whenever there is a request to trace back the subscriber's private IP address, access the right file based on the timestamp provided. The file will have the public IP and port range to which the specified port belongs. The private IP address in that row will help identify the subscriber.

Considerations and Limitations of Predefined NAT

The considerations and the limitations of the predefined mode for NAT 44 are as follows:

- You can configure the predefined mode for each of the inside VRF instance.
- A new parameter, private address range, has been added to the NAT 44 configuration for the predefined mode. You can specify a minimum of one private address range to a maximum of eight private address ranges. Ensure that you specify atleast one private address range because the available public addresses and the associated ports are mapped to the private addreses specified in this range. If the incoming packet has an address that is outside the private address range, then the packet is discarded. Ensure that the sum of all addresses should not exceed one million across all predefined mode-enabled VRFs.

- The Bulk Port Allocation configuration is not available in the predefined mode. If you try to configure Bulk Port Allocation on an inside VRF that has the predefined mode enabled, the configuration is rejected during verification.
- The port-preservation option is not available in the predefined mode.
- The global port limit parameter is not available for the predefined mode. Even though you will be allowed to configure the global port limit, the inside VRF, which has predefined mode enabled, ignores that port limit and uses the port limit configured by the algorithm.
- If you turn the predefined mode on or off for an inside VRF during the active translations, all the translations on that VRF are deleted.
- If a request for configuring static port on a private address that is not in the address range is made, the request is rejected.
- Ensure that you configure NetFlow or syslog only if it is very much required.
- Any configuration change that results in changes in mapping deletes the existing translations. Therefore, ensure that you record such configuration changes. You might need this information to trace the port usage by a subscriber.
- Ensure uniform port allocation uniform for all subscribers.

Configuring the Predefined Mode for NAT44

Perform these tasks to configure the predefined mode for NAT44.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44** *nat1*
4. **inside-vrf** *vrf-name*
5. **map address-pool** *address/prefix*
6. **nat-mode**
7. **predefined** *ipaddress/prefix*
8. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**service cgn** *instance-name* **Example:**	Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1` `RP/0/RP0/CPU0:router(config-cgn)#`	
Step 3	**service-type nat44** *nat1* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures an inside VRF named *insidevrf1* and enters CGv6 inside VRF configuration mode.

Step 5	**map address-pool** *address/prefix* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf)# map address-pool 10.10.0.0/16`	Maps an inside VRF to an outside VRF and address pool mapping.
Step 6	**nat-mode** **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf)# nat-mode`	Specifies the predefined mode for NAT44.

Step 7	**predefined** *ipaddress/prefix*
	Example:
	`RP/0/RP1/CPU0:router(config-cgn-invrf-natmode)# predefined private-pool 192.1.106.0/24`
	`RP/0/RP1/CPU0:router(config-cgn-invrf-natmode)# predefined private-pool 192.1.107.0/26`
	`RP/0/RP1/CPU0:router(config-cgn-invrf-natmode)# predefined private-pool 192.1.107.128/26`
		Specifies the private address range for the predefined mode. You can specify a minimum of one address range to eight addess ranges.

Step 8	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-natmode)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-natmode)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring NAT64 over VSM

Perform the following tasks to configure NAT64 on VSM:

1. Configuring a NAT64 Instance, on page 31
2. Configuring the Application Service Virtual Interface (NAT64), on page 32
3. Configuring an Inside and Outside Address Pool Map (NAT64), on page 34

Configuring a NAT64 Instance

Perform this task to configure a NAT64 instance.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-location preferred-active** *VSM location*
4. **service-type nat64 stateful** *instance-name*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 NAT64 application and enters CGv6 configuration mode.

Step 3	**service-location preferred-active** *VSM location* **Example:**	Configures the NAT preferred active VSM location.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-cgn)# service-location preferred-active 0/3/CPU0`	
Step 4	**service-type nat64 stateful** *instance-name* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat64 stateful nat64-inst`	Configures the service type keyword definition for CGv6 Stateful NAT64 application.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Application Service Virtual Interface (NAT64)

Perform this task to configure the application service virtual interface (SVI) to forward data traffic.

SUMMARY STEPS

1. **configure**
2. **interface ServiceApp** *value*
3. **vrf** *vrf-name*
4. **ipv6 address***address*
5. **service cgn** *instance-name* **service-type nat64 stateful**
6. **commit**
7. **interface ServiceApp** *value*
8. **vrf** *vrf-name*
9. **ipv4 address** *address*
10. **service cgn** *instance-name* **service-type nat64 stateful**
11. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**interface ServiceApp** *value*
	Example:
	`RP/0/RP0/CPU0:router(config)# interface ServiceApp 1`
	`RP/0/RP0/CPU0:router(config-if)#`
		Configures the application SVI as 1 and enters interface configuration mode.

Step 3	**vrf** *vrf-name* **Example:** `RP/0/RP0/CPU0:router(config-if)# vrf insidevrf1`	Configures the VPN routing and forwarding (VRF) for the Service Application interface
Step 4	**ipv6 address***address* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv6 address 5001::5555/96`	Configures the IPv6 address.

Step 5	**service cgn** *instance-name* **service-type nat64 stateful**
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1 service-type nat64 stateful`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)#`
		Configures the instance named cgn1 and the service type keyword definition for CGv6 Stateful NAT64 application.

Step 6	**commit** **Example:** `RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)# commit`	Saves configuration changes.

Step 7	**interface ServiceApp** *value*
	Example:
	`RP/0/RP0/CPU0:router(config)# interface ServiceApp 2`
	`RP/0/RP0/CPU0:router(config-if)#`
		Configures the application SVI as 2 and enters interface configuration mode.

Step 8	**vrf** *vrf-name* **Example:** `RP/0/RP0/CPU0:router(config-if)# vrf outsidevrf1`	Configures the VPN routing and forwarding (VRF) for the Service Application interface
Step 9	**ipv4 address** *address* **Example:**	Configures the IPv4 address.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# ipv4 address 211.1.1.1 255.255.255.0`	

Step 10	**service cgn** *instance-name* **service-type nat64 stateful**
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1 service-type nat64 stateful`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)#`
		Configures the instance named cgn1 and the service type keyword definition for CGv6 Stateful NAT64 application.

Step 11	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring an Inside and Outside Address Pool Map (NAT64)

Perform this task to configure an inside and outside address pool map with the following scenarios:

- The designated address pool is used for CNAT.
- One inside VRF is mapped to only one outside VRF.
- Multiple non-overlapping address pools (up to a maximum of eight) can be used in a specified outside VRF mapped to different inside VRF.
- Max Outside public pool per CGSE/CGN instance is 64 K or 65536 addresses. That is, if a /16 address pool is mapped, then we cannot map any other pool to that particular CGSE.
- Multiple inside vrf cannot be mapped to same outside address pool.
- While Mapping Outside Pool Minimum value for prefix is 16 and maximum value is 26.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat64 stateful** *instance-name*
4. **ipv6-prefix** *address/prefix*
5. **ipv4-address-pool** *address/prefix*
6. **dynamic-port-range start** *port-number*
7. **address-family ipv4**
8. **interface ServiceApp** *value*
9. **address-family ipv6**
10. **interface ServiceApp** *value*
11. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**service cgn** *instance-name* **Example:** `RP/0/RP0/CPU0:router(config)#service cgn cgn1`	Configures the instance named cgn1 for the CGv6 application and enters the CGv6 configuration mode.
Step 3	**service-type nat64 stateful** *instance-name* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat64 stateful nat64_1`	Configures the service type keyword definition for CGv6 Stateful NAT64 application.
Step 4	**ipv6-prefix** *address/prefix* **Example:** `RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)# ipv6-prefix 3301:db8:1::/96`	Configures the IPv6 prefix that is used to convert destination IPv6 address to an external destination IPv4 address.
Step 5	**ipv4-address-pool** *address/prefix* **Example:** `RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)# ipv4 address-pool 62.1.0.0/18`	Defines the IPv4 address pool.
Step 6	**dynamic-port-range start** *port-number* **Example:** `RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)#dynamic-port-range start 1`	Configures the port range from 1 to 65535.

	Command or Action	Purpose
Step 7	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)# address-family ipv4`	Enters the address family IPv4 configuration mode.
Step 8	**interface ServiceApp** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-nat64-stful-afi)# interface ServiceApp2`	Specifies the ServiceApp on which IPv4 traffic enters and leaves.
Step 9	**address-family ipv6** **Example:** `RP/0/RP0/CPU0:router(config-cgn-nat64-stful-afi)# address-family ipv6`	Enters the address family IPv6 configuration mode.
Step 10	**interface ServiceApp** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-nat64-stful-afi)# interface ServiceApp1`	Specifies the ServiceApp on which IPv6 traffic enters and leaves.
Step 11	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cgn-nat64-stful-afi)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: *Uncommitted changes found, commit them before exiting (yes/no/cancel)?* *[cancel]:* • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Policy Functions

Configuring Port Limit per Subscriber

Perform this task to restrict the number of ports used by an IPv6 address.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat64 stateful** *instance-name*
4. **portlimit** *value*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type nat64 stateful** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type nat64 stateful nat64-inst`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)#`
		Configures the service type keyword definition for CGv6 Stateful NAT64 application.

Step 4	**portlimit** *value*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)#portlimit 66`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)`
		Configures a value to restrict the number of ports used by an IPv6 address.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:

	Command or Action	Purpose
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Timeout Value for ICMP, TCP and UDP Sessions

Perform this task to configure the timeout value for ICMP, TCP or UDP sessions for a Dual Stack Lite (DS Lite) instance:

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type ds-lite** *instance-name*
4. **protocol tcp session {active | initial} timeout** *value* or **protocol {icmp | udp} timeout** ***value***
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type ds-lite** *instance-name* **Example:**	Configures the service type keyword definition for CGv6 DS-Lite application.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-cgn)# service-type ds-lite ds-lite-inst` `RP/0/RP0/CPU0:router(config-cgn-ds-lite)#`	

Step 4	**protocol tcp session {active | initial} timeout** *value* or **protocol {icmp | udp} timeout** ***value***
	 Example:
	 `RP/0/RP0/CPU0:router(config-cgn-ds-lite)#protocol tcp session active timeout 90`
	`or`
	`protocol icmp timeout 90`
	`RP/0/RP0/CPU0:router(config-cgn-ds-lite)`
		Configures the initial and active session timeout values for TCP.
		Configures the timeout value in seconds for ICMP and UDP.

Step 5	**end** or **commit**
	 Example:
	 `RP/0/RP0/CPU0:router(config-cgn-ds-lite)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-ds-lite)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

FTP-ALG

CGN supports both passive and active FTP. FTP clients are supported with inside (private) address and servers with outside (public) addresses. Passive FTP is provided by the basic NAT function. Active FTP is used with the ALG.

RTSP-ALG

CGN supports the Real Time Streaming Protocol (RTSP), an application-level protocol for control over the delivery of data with real-time properties. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. Sources of data can include both live data feeds and stored clips.

PPTP-ALG

PPTP is a network protocol that enables secure transfer of data from a remote client to a private enterprise server by creating a Virtual Private Network (VPN). It is used to provide IP security at the network layer. PPTP uses a control channel over TCP and a GRE tunnel operating to encapsulate PPP packets.

PPTP-ALG is a CGN solution that allows traffic from all clients through a single PPTP tunnel.

A PPTP tunnel is instantiated on the TCP port. This TCP connection is then used to initiate and manage a second GRE tunnel to the same peer.

PPTP uses an access controller and network server to establish a connection.

PPTP Access Controller (PAC)

A device attached to one or more PSTN or ISDN lines capable of PPP operation and handling the PPTP protocol. It terminates the PPTP tunnel and provides VPN connectivity to a remote client.

PPTP Network Server (PNS)

A device which provides the interface between the Point-to-Point Protocol (encapsulated in the PPTP protocol) and a LAN or WAN. The PNS uses the PPTP protocol to support tunneling between a PPTP PAC and the PNS. It requests to establish a VPN connectivity using PPTP tunnel.

Control Connection

A control connection is established between a PAC and a PNS for TCP.

Tunnel

A tunnel carries GRE encapsulated PPP datagrams between a PAC and a PNS

Note Active FTP, PPTP ALG, and RTSP ALG are supported on NAT44 applications. Active FTP and RTSP ALG are supported on DS-Lite applications.

TCP Maximum Segment Size Adjustment

When a host initiates a TCP session with a server, the host negotiates the IP segment size by using the maximum segment size (MSS) option. The value of the MSS option is determined by the maximum transmission unit (MTU) that is configured on the host.

Static Port Forwarding

Static port forwarding helps in associating a private IP address and port with a statically allocated public IP and port. After you have configured static port forwarding, this association remains intact and does not get removed due to timeouts until the CGSE is rebooted. In case of redundant CGSE cards, it remains intact until both of the CGSEs are reloaded together or the router is reloaded. There are remote chances that after a reboot, this association might change. This feature helps in cases where server applications running on the private network needs access from public internet.

Note NAT64 on VSM does not support static port forwarding.

Configuring Dynamic Port Range

Perform this task to configure a dynamic port range.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat64 stateful** *instance-name*
4. **dynamic-port-range start** *port-number*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type nat64 stateful** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type nat64 stateful nat64-inst`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)`
		Configures the service type keyword definition for CGv6 Stateful NAT64 application.

Step 4	**dynamic-port-range start** *port-number*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)#dynamic-port-range start 66`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)`
		Configures the port range from 1 to 65535.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

(Only NAT44) Configuring One-to-One Mapping

Perform this task to configure one-to-one mapping for private addresses.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **map ip one-to-one**
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGv6 NAT44 application.

	Command or Action	Purpose

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGv6 instance named cgn1 and enters CGv6 inside VRF configuration mode.

Step 5	**map ip one-to-one** **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf)# map ip one-to-one`	Configures one-to-one mapping for a CGv6 NAT44 instance.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-afi)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-afi)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

(Only NAT44) Configuring TCP Sequence Check

Perform the following steps for checking the sequence numbers of the packets in a TCP session:

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **firewall protocol tcp**
6. **sequence-check**
7. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGN instance named cgn1 and enters CGN inside VRF configuration mode.

Step 5	**firewall protocol tcp**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# protocol tcp`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-proto)#`
		Enters the firewall mode and the protocol tcp submode.

Step 6	**sequence-check** **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf)# sequence-check 10`	Enables checking of the sequence numbers. The optional **diff-window** keyword allows user to configure a value equal to the difference between the expected and received sequence numbers. The range for this value is 0 to 1,073,725,440.

Step 7	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-ivrf-sport-inside)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-ivrf-sport-inside)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

(Only NAT44) Enabling Address and Port-Dependent Filtering

Perform the following steps to enable address and port-dependent filtering in a NAT44 configuration.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44** *instance-name*
4. **inside-vrf** *instance-name*
5. **filter-policy ignore-port**
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type nat44** *instance-name* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGv6 NAT44 application.

	Command or Action	Purpose

Step 4	**inside-vrf** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures an inside VRF named insidevrf1 and enters CGv6 inside VRF configuration mode.

Step 5	**filter-policy ignore-port**
	Example:
	To enable address and port-dependent filtering
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# filter-policy`
	To enable address and port-dependent filtering when the port is not checked:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# filter-policy ignore-port`
		Enables the address and port-dependent filtering. The optional **ignore-port** keyword is used to disable the port-dependent filtering.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

(Only NAT44) Configuring NAT0 Mode

Perform these tasks to configure the NAT0 mode for NAT44.

SUMMARY STEPS

1. **configure**

2. **service cgn** *instance-name*
3. **service-type nat44** *nat1*
4. **inside-vrf** *vrf-name*
5. **map** [**outside-vrf** *outside-vrf-name*] **address-pool** *address/prefix*
6. **nat-mode** *no-nat*
7. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44** *nat1* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures an inside VRF named *insidevrf1* and enters CGv6 inside VRF configuration mode.

Step 5	**map** [**outside-vrf** *outside-vrf-name*] **address-pool** *address/prefix* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf)# map outside-vrf ovrf outsideServiceApp ServiceApp2 address-pool 0.0.0.0/0`	Configures an inside VRF to an outside VRF and address pool mapping.
Step 6	**nat-mode** *no-nat* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf)# nat-mode no-nat`	Specifies the NAT0 or NAT bypass mode for NAT44.
Step 7	**end** or **commit** **Example:**	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose

	`RP/0/RP0/CPU0:router(config-cgn-invrf-natmode)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-invrf-natmode)# commit`
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

(Only NAT44) Configuring the Static Destination Address Translation

Perform the following steps to configure static destination address translation:

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **map** [**outside-vrf** *outside-vrf-name*] **address-pool** *address/prefix*
6. **static-mapping-file direction i2o-dst** *path of the .csv file*
7. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**service cgn** *instance-name* **Example:**	Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1` `RP/0/RP0/CPU0:router(config-cgn)#`	
Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGN NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures an inside VRF named insidevrf1 and enters CGN inside VRF configuration mode.

Step 5	**map** [**outside-vrf** *outside-vrf-name*] **address-pool** *address/prefix*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# map outside-vrf outside vrf1 address-pool 10.10.0.0/16`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# map address-pool 100.1.0.0/16`
		Configures an inside VRF to an outside VRF and address pool mapping.

Step 6	**static-mapping-file direction i2o-dst** *path of the .csv file* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-afi)# static-mapping-file direction i2o-dst /disk0:/30K.CSV`	Configures static destination address translation using the static mapping configuration file (.csv). In this example, the file is stored in disk0: location.

Step 7	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-afi)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-afi)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

(Only NAT44) Configuring Multiple NetFlow Servers

Perform this task to configure multiple NetFlow servers. Repeat the same task to configure multiple servers.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **external-logging netflow**
6. **server**
7. **address** *address* **port** *number*
8. **ath-mtu** *value*
9. **refresh-rate** *value*
10. **timeout** *value*
11. **session-logging**
12. **end** or **commit**
13.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGv6 NAT44 application.

	Command or Action	Purpose

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-nat44-invrf)#`
		Configures the inside VRF for the CGv6 instance named cgn1 and enters CGv6 inside VRF configuration mode.

Step 5	**external-logging netflow**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging netflow version 9`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the NAT44 instance.

Step 6	**server**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the netflow-v9 based external-logging facility.

Step 7	**address** *address* **port** *number* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# address 2.3.4.5 port 45`	Configures the IPv4 address and port number 45 to log Netflow entries for the NAT table.
Step 8	**ath-mtu** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# path-mtu 2900`	Configures the path MTU with the value of 2900 for the netflowv9-based external-logging facility.
Step 9	**refresh-rate** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# refresh-rate 50`	Configures the refresh rate value of 50 to log Netflow-based external logging information for an inside VRF.
Step 10	**timeout** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# timeout 50`	Configures the timeout value of 50 for Netflow logging of NAT table entries for an inside VRF.
Step 11	**session-logging** **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# session-logging`	Configures the session logging for a NAT44 instance.
Step 12	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose
	or RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit	*Uncommitted changes found, commit them before exiting (yes/no/cancel)?* *[cancel]:* • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Step 13		To configure a second server, repeat the steps from 7 to 11.

(Only NAT64) Configuring External Logging for NetFlow Servers

Perform this task to configure external logging for NetFlow servers.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-location preferred-active** *node-id*
4. **service-type nat64 stateful** *name*
5. **external-logging netflow**
6. **server**
7. **address** *address* **port** *number*
8. **ath-mtu** *value*
9. **refresh-rate** *value*
10. **timeout** *value*
11. **session-logging**
12. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-location preferred-active** *node-id* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-location preferred-active 0/1/CPU01`	Specifies the location in which the application starts.
Step 4	**service-type nat64 stateful** *name* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat64 stateful nat64_1`	Configures the service type keyword and the name for CGv6 NAT64 application.

Step 5	**external-logging netflow**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging netflow version 9`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the NAT64 instance.

Step 6	**server**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv6 address and port for the server that is used for the netflow-v9 based external-logging facility.

Step 7	**address** *address* **port** *number* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# address 2.3.4.5 port 614`	Configures the IPv6 address and port number 614 to log Netflow entries for the NAT table.
Step 8	**ath-mtu** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# path-mtu 2900`	Configures the path MTU with the value of 2900 for the netflowv9-based external-logging facility.

	Command or Action	Purpose
Step 9	**refresh-rate** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# refresh-rate 50`	Configures the refresh rate value of 50 to log Netflow-based external logging information for an inside VRF.
Step 10	**timeout** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# timeout 50`	Configures the timeout value of 50 for Netflow logging of NAT table entries for an inside VRF.
Step 11	**session-logging** **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# session-logging`	Configures the session logging for a NAT64 instance.

Step 12	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

(Only NAT44) Configuring Multiple Syslog Servers

Perform this task to configure an additional Syslog server. Repeat the same task to configure multiple servers.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**

4. **inside-vrf** *vrf-name*
5. **external-logging syslog**
6. **server**
7. **address***address***port***number*
8. **path-mtu***value*
9. **refresh-rate** *value*
10. **timeout** *value*
11. **session-logging**
12. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGv6 NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-nat44-invrf)#`
		Configures the inside VRF for the CGv6 instance named cgn1 and enters CGv6 inside VRF configuration mode.

Step 5	**external-logging syslog**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging syslog`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the NAT44 instance.

Step 6	**server**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the netflow-v9 based external-logging facility.

	Command or Action	Purpose
Step 7	**address***address***port***number* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# address 2.3.4.5 port 45`	Configures the IPv4 address and port number 45 to log Netflow entries for the NAT table.
Step 8	**path-mtu***value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# path-mtu 200`	Configures the path MTU with the value of 200 for the syslog-based external-logging facility.
Step 9	**refresh-rate** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# refresh-rate 50`	Configures the refresh rate value of 50 to log Netflow-based external logging information for an inside VRF.
Step 10	**timeout** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# timeout 50`	Configures the timeout value of 50 for Netflow logging of NAT table entries for an inside VRF.
Step 11	**session-logging** **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# session-logging`	Configures the session logging for a NAT44 instance.

Step 12	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Example

(Only NAT44) Configuring External Logging for the NAT Table Entries

Perform the following to configure external logging for NAT table entries.

Configuring the Server Address and Port for Netflow Logging

Perform this task to configure the server address and port to log network address translation (NAT) table entries for Netflow logging.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **external-logging netflowv9**
6. **server**
7. **address** *address* **port** *number*
8. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGN instance named cgn1 and enters CGN inside VRF configuration mode.

	Command or Action	Purpose

Step 5	**external-logging netflowv9**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging netflowv9`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the CGN instance named cgn1 and enters CGN inside VRF address family external logging configuration mode.

Step 6	**server**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the netflowv9-based external-logging facility and enters CGN inside VRF address family external logging server configuration mode.

Step 7	**address** *address* **port** *number* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# address 2.3.4.5 port 45`	Configures the IPv4 address and port number 45 to log Netflow entries for the NAT table.

Step 8	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Path Maximum Transmission Unit for Netflow Logging

Perform this task to configure the path maximum transmission unit (MTU) for the netflowv9-based external-logging facility for the inside VRF.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **external-logging netflowv9**
6. **server**
7. **path-mtu** *value*
8. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGN instance named cgn1 and enters CGN inside VRF configuration mode.

Step 5	**external-logging netflowv9**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging netflowv9`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the CGN instance named cgn1 and enters CGN inside VRF address family external logging configuration mode.

Step 6	**server**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the netflowv9-based external-logging facility and enters CGN inside VRF address family external logging server configuration mode.

	Command or Action	Purpose
Step 7	**path-mtu** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# path-mtu 2900`	Configures the path MTU with the value of 2900 for the netflowv9-based external-logging facility.

Step 8	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Refresh Rate for Netflow Logging

Perform this task to configure the refresh rate at which the Netflow-v9 logging templates are refreshed or resent to the Netflow-v9 logging server.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **external-logging netflowv9**
6. **server**
7. **refresh-rate** *value*
8. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGN instance named cgn1 and enters CGN inside VRF configuration mode.

Step 5	**external-logging netflowv9**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging netflowv9`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the CGN instance named cgn1 and enters CGN inside VRF address family external logging configuration mode.

Step 6	**server**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the netflowv9-based external-logging facility and enters CGN inside VRF address family external logging server configuration mode.

Step 7	**refresh-rate** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# refresh-rate 50`	Configures the refresh rate value of 50 to log Netflow-based external logging information for an inside VRF.

Step 8	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`
	`or`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit`	*[cancel]:* • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Timeout for Netflow Logging

Perform this task to configure the frequency in minutes at which the Netflow-V9 logging templates are to be sent to the Netflow-v9 logging server.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **external-logging netflowv9**
6. **server**
7. **timeout***value*
8. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

	Command or Action	Purpose
Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGN instance named cgn1 and enters CGN inside VRF configuration mode.

Step 5	**external-logging netflowv9**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging netflowv9`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the CGN instance named cgn1 and enters CGN inside VRF address family external logging configuration mode.

Step 6	**server**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the netflowv9-based external-logging facility and enters CGN inside VRF address family external logging server configuration mode.

Step 7	**timeout***value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# timeout 50`	Configures the timeout value of 50 for Netflow logging of NAT table entries for an inside VRF.

Step 8	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

(Only NAT44) Syslog Logging

Perform the following tasks to configure Syslog Logging for NAT table entries.

Configuring the Server Address and Port for Syslog Logging

Perform this task to configure the server address and port to log DS-Lite entries for Syslog logging.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type ds-lite** *instance_name*
4. **external-logging syslog**
5. **server**
6. **address***address***port***number*
7. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type ds-lite** *instance_name* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type ds-lite ds-lite1`	Configures the service type keyword definition for the DS-Lite application.

	Command or Action	Purpose

Step 4	**external-logging syslog**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-ds-lite)#external-logging syslog`
	`RP/0/RP0/CPU0:router(config-cgn-ds-lite-extlog)#`
		Configures the external-logging facility for the CGv6 instance named cgn1 and enters CGv6 external logging configuration mode.

Step 5	**server**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-ds-lite-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-ds-lite-extlogserver)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the syslog-based external-logging facility and enters CGv6 external logging server configuration mode.

Step 6	**address***address***port***number* **Example:** `RP/0/RP0/CPU0:router(config-cgn-ds-lite-extlogserver)#address 2.3.4.5 port 45`	Configures the IPv4 address and port number 45 to log Netflow entries.

Step 7	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-ds-lite-extlogserver)#end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-ds-lite-extlogserver)#commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Host-Name for Syslog Logging

Perform this task to configure the host name to be filled in the Netflow header for the syslog logging.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*

3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **external-logging syslog**
6. **server**
7. **host-name***name*
8. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGv6 NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGv6 instance named cgn1 and enters CGv6 inside VRF configuration mode.

Step 5	**external-logging syslog**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging syslog`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the CGv6 instance named cgn1 and enters CGv6 inside VRF address family external logging configuration mode.

Step 6	**server**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the syslog-based external-logging facility and enters CGv6 inside VRF address family external logging server configuration mode.

Step 7	**host-name***name* **Example:**	Configures the host name for the syslog-based external-logging facility.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# host-name host1`	

Step 8	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Path Maximum Transmission Unit for Syslog Logging

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat44 nat1**
4. **inside-vrf** *vrf-name*
5. **external-logging syslog**
6. **server**
7. **path-mtu***value*
8. **end** or **commit**

DETAILED STEPS

Procedure

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

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# configure`	

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type nat44 nat1** **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for CGv6 NAT44 application.

Step 4	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf insidevrf1`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures the inside VRF for the CGv6 instance named cgn1 and enters CGv6 inside VRF configuration mode.

Step 5	**external-logging syslog**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# external-logging syslog`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)#`
		Configures the external-logging facility for the CGv6 instance named cgn1 and enters CGv6 inside VRF address family external logging configuration mode.

Step 6	**server**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog)# server`
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)#`
		Configures the logging server information for the IPv4 address and port for the server that is used for the syslog-based external-logging facility and enters CGv6 inside VRF address family external logging server configuration mode.

Step 7	**path-mtu***value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# path-mtu 200`	Configures the path MTU with the value of 200 for the syslog-based external-logging facility.

Step 8	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-invrf-af-extlog-server)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

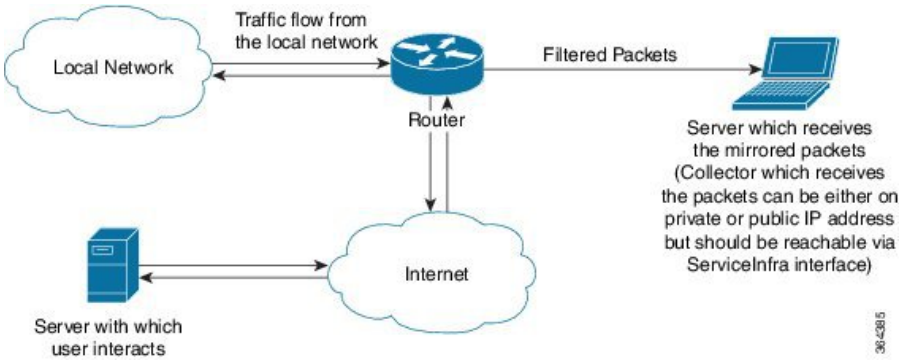
Traffic Flow Mirroring

Traffic flow mirroring is a solution which enables you to monitor the incoming and outgoing traffic on the VSM module (of ASR9K) running a CGN instance. This solution helps you to debug and analyze packets for issues pertaining to NAT-ing (NAT44). The traffic is filtered based on a set of particular parameters, which can be set by the user. The packets, collected, are encapsulated in a GRE envelope and sent to a pre-configured collector like a UNIX system, laptop, etc. This envelope contains a field, which provides information about the type of packet whether the packets are In2Out packet, Out2In packet, pre-NAT, post-NAT, or dropped, analyzing this field information, the issues pertaining to NAT can be debugged.

Salient Features:

- Any packets dropped will be mirrored.
- The packets are filtered based on destination address; and refined further based on port number, protocol, and IP addresses of the subscriber devices that are mirrored.
- Mirroring of up to 16 VRFs is supported when the destination address filter is configured. There is no limit on the number of VRFs supported when the mirroring is enabled for only the dropped packets.

Figure 1: Traffic Flow Mirroring Topology



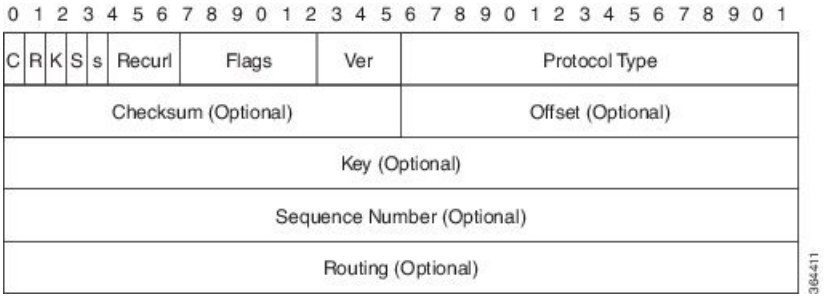
If the packets are filtered based on the destination IP address, then destination IP address is a mandatory field for the solution whereas a few of the fields like protocol used, destination port, private source prefix, etc. are optional.

Note Mirroring occurs only for packets that are intercepted after the feature is turned on.

Mirrored Packet Data Interpretation

The packets received at the collector have the original packet as the payload encapsulated in a GRE header. A typical GRE header is as shown in the following figure.

Figure 2: GRE Header



The KEY field in the GRE header contains the value. The following table lists the values and the description associated with those values.

Table 2: List of KEY field values and their descriptions:

Value	Description
1	In to Out direction, pre-nat packet
2	Out to In direction, pre-nat packet
3	In to Out direction, post-nat packet
4	Out to In direction, post-nat packet
5	Dropped In to Out TCP, PPTP control message packet.
6	Dropped In to Out TCP Fragment packet. (Received non-first fragment.)
7	Dropped In to Out TCP packet. (Failed to create new NAT entry.)
8	Dropped In to Out TCP packet due to no session entry.
9	Dropped In to Out TCP packet. (Source port is zero)
10	Dropped In to Out TCP packet. (None sync drop)
11	Dropped In to Out TCP packet (Session creation fail)
12	Dropped In to Out TCP packet with TTL <= 1. (No ICMP generated due to throttling)
13	Dropped packet as ICMP is sent for first fragment only.

14	Dropped packet due to Invalid ICMP error code.
15	Dropped In to Out due ICMP error packet with TTL <= 1.
16	Dropped In to Out ICMP packet due to no NAT entry.
17	Dropped Out to In ICMP packet (ipv4 packet too large for the tunnel)
18	Dropped Out to In ICMP packet due to no NAT entry.
19	Dropped In to Out ICMP packet due to no session.
20	Dropped In to Out ICMP packet with TTL <= 1. (No ICMP generated due to throttling)
21	Dropped In to Out ICMP query packet due to no NAT entry.
22	Dropped Out to In ICMP query. (No NAT entry)
23	Dropped Out to In ICMP query packet due to end point filtering. (EDF is enabled).
24	Dropped Out to In ICMP query packet, could not generate ICMP packet due to throttling.
25	Dropped Out to In ICMP packet due to no session.
26	Dropped Out to In ICMP packet due to no NAT entry.
27	Dropped port control protocol (PCP) packet, as it couldn't be handled.
28	Dropped In to Out PPTP packet (PPTP not configured)
29	Dropped In to Out PPTP packet with TTL <= 1 (No ICMP generated due to throttling)
30	Dropped Out to In PPTP packet (PPTP not configured)
31	Dropped Out to In PPTP fragment packet (No NAT entry)
32	Dropped Out to In PPTP packet (No NAT entry)
33	Dropped Out to In PPTP packet with TTL <= 1. (No ICMP generated due to throttling.)
34	Dropped In to Out UDP packet (Has no available ports)
35	Dropped In to Out UDP packet (UDP port value of 0).
36	Dropped In to Out UDP packet (No configuration available).
37	Dropped In to Out UDP packet (No ICMP message generated).
38	Dropped In to Out UDP packet (Create session failed).
39	Dropped In to Out UDP packet (VRF not in run state)

40	Dropped In to Out UDP packet (Port limit exceeded)
41	Dropped In to Out UDP packet with TTL <= 1. (No ICMP generated due to throttling.)
42	Dropped In to Out UDP packet (No direct port available).
43	Dropped Out to In UDP packet (No NAT entry).
44	Dropped Out to In UDP packet due to end point filtering. (EDF is enabled)
45	Dropped Out to In UDP packet (No NAT entry).
46	Dropped Out to In UDP packet (Create session DB failed or Session limit exceeded.)
47	Dropped Out to In UDP packet as it is too large for tunneling. **Note** ICMP not generated due to throttling.
48	Dropped Out to In UDP packet (Create session failed.)
49	Dropped Out to In UDP fragment packet (No NAT entry).
50	Not used
51	Dropped Out to In Error fragment packet.
52	Dropped Out to In unsupported protocol Fragment packet.
53	Dropped Out to In TCP packet (PPTP control message dropped.)
54	Dropped Out to In TCP packet (No NAT entry)
55	Dropped Out to In TCP packet (First fragment packet drop)
56	Dropped Out to In TCP due to end point filtering. (EDF is enabled.)
57	Dropped Out to In UDP packet as it is too large for tunneling. **Note** ICMP not generated due to throttling.
58	Dropped Out to In TCP packet. (Create session failed.)
59	Dropped Out to in TCP fragment packet (No NAT entry)
60	Dropped Out to in TCP packet (SYN or RST flags not set for TCP session to be established.)
61	Dropped Out to in TCP packet (Sequence mismatch)
62	Dropped Out to In TCP packet with TTL <= 1. (No ICMP generated due to throttling.)

Limitations and Assumptions

The following are a few of the assumptions and limitations of the traffic flow mirroring solution:

- At any given point in time, only one traffic flow mirroring per inside-vrf is allowed.
- If the collector IP address is not configured, the traffic packet mirroring is blocked. In case the collector IP address is not reachable, the mirrored packets are dropped.
- If the protocol is not provided, both TCP and UDP packets are mirrored.
- If the port number is not mentioned, the traffic flowing through all the destination ports are mirrored.
- If a private source IP address is not configured, the mirroring is performed for all subscribers of the VRF, that is listed. This can reduce the performance of VSM and also lead to choking the collector. It is advisable to configure as many parameters as possible to filter and mirror only the required packets.
- Performance figures of VSM are not guaranteed when traffic mirroring is on.
- Traffic flow mirroring solution assumes that the collector is reachable to the router in the default VRF. The router does not attempts to ping or get acknowledgments to ascertain if the collector is receiving the packets.

Configuring Mirroring Using Destination Address Filter and Collector IP Address

Perform this task to configure mirroring the traffic packets using a destination address filter and collector IP address.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-location preferred-active** *node-id*
4. **service-type nat44 nat1** *instance*
5. **inside-vrf** *vrf-name*
6. **mirror-packets**
7. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-location preferred-active** *node-id* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-location preferred-active 0/1/CPU0`	Specifies the global command applied per CGN instance. It initiates the particular instance of the CGN application on the active and standby locations.
Step 4	**service-type nat44 nat1** *instance* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for the NAT44 NAT1 application.

Step 5	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf BLR_BTM`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures an inside VRF named BLR_BTM and enters CGN inside VRF configuration mode.

Step 6	**mirror-packets**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# mirror-packets`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# destination-ipv4-address 201.22.3.45`
	`!`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# collector-ipv4-address 187.2.3.55`
	`!`
	`!`
		Filters the traffic such that the packets are mirrored onto the provided destination collector IP address.

Step 7	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Example

The following example shows how to filter and configure data packets to be mirrored onto a collector with the destination IP address and the collector IP address provided.

```
service cgn cgn1
 service-location preferred-active 0/1/CPU0
 service-type nat44 nat1
  inside-vrf BLR_BTM
   mirror-packets
    destination-ipv4-address 201.22.3.45
    !
    collector-ipv4-address 187.2.3.55
   !
  !
 !
!
```

Configuring Mirroring Using Destination Address, Port Number, Protocol Type, Source-Prefix Filters, and Collector IP Address

Perform this task to configure mirroring the traffic packets using a destination address, port number, protocol type, source-prefix filter and collector IP address.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-location preferred-active** *node-id*
4. **service-type nat44 nat1** *instance*
5. **inside-vrf** *vrf-name*
6. **mirror-packets**
7. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-location preferred-active** *node-id* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-location preferred-active 0/1/CPU0`	Specifies the global command applied per CGN instance. It initiates the particular instance of the CGN application on the active and standby locations.
Step 4	**service-type nat44 nat1** *instance* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for the NAT44 NAT1 application.

Step 5	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf BLR_BTM3`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures an inside VRF named BLR_BTM3 and enters CGN inside VRF configuration mode.

Step 6	**mirror-packets**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
	`mirror-packets`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
	`destination-ipv4-address 201.22.3.45`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
	`protocol-type tcp udp`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# port 4002`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
	`source-prefix 100.1.1.252/30`
	`!`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
	`collector-ipv4-address 187.2.3.5`
	`!`
	`!`
		Configures the traffic packets to be mirrored onto the provided destination collector IP address.

	Command or Action	Purpose

Step 7	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Example

The following example shows how to filter and configure packets to be mirrored onto a collector with the destination details like the IP address, protocol type, port number, source-prefix filter, and the collector IP address.

```
service cgn cgn1
 service-location preferred-active 0/1/CPU0
 service-type nat44 nat1
  inside-vrf BLR_BTM3
   mirror-packets
    destination-ipv4-address 201.22.3.45
     protocol-type tcp udp
     port 4002
     source-prefix 100.1.1.252/30
    !
    collector-ipv4-address 187.2.4.5
   !
  !
 !
!
```

Configuring Mirroring for Dropped Packets Using Collector IP Address

Perform this task to configure mirroring the dropped traffic packets using collector IP address.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-location preferred-active** *node-id*
4. **service-type nat44 nat1** *instance*
5. **inside-vrf** *vrf-name*
6. **mirror-packets**
7. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGN application and enters CGN configuration mode.

Step 3	**service-location preferred-active** *node-id* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-location preferred-active 0/1/CPU0`	Specifies the global command applied per CGN instance. It initiates the particular instance of the CGN application on the active and standby locations.
Step 4	**service-type nat44 nat1** *instance* **Example:** `RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`	Configures the service type keyword definition for the NAT44 NAT1 application.

Step 5	**inside-vrf** *vrf-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type nat44 nat1`
	`RP/0/RP0/CPU0:router(config-cgn-nat44)# inside-vrf BLR_BTM3`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
		Configures an inside VRF named BLR_BTM3 and enters CGN inside VRF configuration mode.

Step 6	**mirror-packets** **Example:** `RP/0/RP0/CPU0:router(config-cgn-invrf)# mirror-packets`	Configures the dropped traffic packets to be mirrored onto the provided destination collector IP address.

	Command or Action	Purpose

	`RP/0/RP0/CPU0:router(config-cgn-invrf)# all-drops`
	`!`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)#`
	`collector-ipv4-address 187.2.3.56`
	`!`
	`!`	

Step 7	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-invrf)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?
		[cancel]:
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Example

The following example shows how to filter and configure dropped traffic packets to be mirrored onto a collector with the IP address provided.

```
service cgn cgn1
 service-location preferred-active 0/1/CPU0
 service-type nat44 nat1
  inside-vrf BLR_BTM2
   mirror-packets
    all-drops
    collector-ipv4-address 187.2.3.56
   !
  !
 !
!
```

Mapping of Address and Port-Encapsulation Mode

Mapping of Address and Port-Encapsulation Mode (MAP-E) is a CGN solution that allows a service provider to enable IPv4 services at IPv6 (customer) sites to which it provides Customer Premise Equipment (CPE). This approach utilizes stateless IPv4-in-IPv6 encapsulation to transit IPv6-enabled network infrastructure. The encapsulation must be supported by the CPE and MAP-E Gateway/Border Relay, which removes the IPv6 encapsulation from IPv4 packets while forwarding them to the Internet. The provider access network can now be on IPv6, while customers see IPv6 and IPv4 service simultaneously.

MAP-E also helps manage IPv4 address exhaustion by keeping the stateful NAT44 on CPE. MAP-E is not supported on any of the VRF interfaces, that is, either IPv4 or IPv6, whereas Map-T is supported with VRF interfaces along with an SMU.

Note

- If the cumulative sum of EA bit value and cpe-domain ipv6 prefix value is more than 64 for an interface, then the traffic is dropped for that interface. EA bit value is calculated as (32- ipv4 prefix) + X), where 32 is a constant and X is the power value of 2 when the sharing ratio is expressed as 2^X.

  Ensure that the EA bit value is less than or equal to 32 and sharing ratio is less than or equal to 256 to avoid any traffic drops.

- You should not modify or delete the existing MAP-E and MAP-T configuration. If you modify the existing configuration, the changes are not reflected in the PBR policy.

  To update the configuration, delete the existing MAP-E and MAP-T instance and add new instance with the required changes.

- Do not configure MAP-E and MAP-T on the same CGv6 instance simultaneously.

Configuring MAP-E

Perform these tasks to configure MAP-E.

Configuring the Application Service Virtual Interface

This section lists the guidelines for selecting service application interfaces for MAP-E.

- Pair ServiceApp<n> with ServiceApp<n+1>, where <n> is an odd integer. This is to ensure that the ServiceApp pairs works with a maximum throughput. For example, ServiceApp1 with ServiceApp2 or ServiceApp3 with ServiceApp4.
- Pair ServiceApp<n> with ServiceApp<n+5> or ServiceApp<n+9>, and so on, where <n> is an odd integer. For example, ServiceApp1 with ServiceApp6, ServiceApp1 with ServiceApp10, ServiceApp3 with ServiceApp8, or ServiceApp3 with ServiceApp12.
- Pair ServiceApp<n> with ServiceApp<n+4>, where <n> is an integer (odd or even integer). For example, ServiceApp1 with ServiceApp5, or ServiceApp2 with ServiceApp6.

Warning Although ServiceApp pairs work, the aggregate throughput for Inside-to-Outside and Outside-to-Inside traffic for the ServiceApp pair is halved.

Caution Do not pair ServiceApp<n> with ServiceApp<n+1>, where <n> is an even integer. When used, Outside-to-Inside traffic is dropped because traffic flows in the incorrect dispatcher and core.

Perform this task to configure the application service virtual interface (SVI) to forward data traffic.

SUMMARY STEPS

1. **configure**
2. **interface ServiceApp** *value*
3. **service cgn** *instance -name* **service-type map-e**
4. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**interface ServiceApp** *value*
	Example:
	`RP/0/RP0/CPU0:router(config)# interface ServiceApp1`
	`ipv4 address 40.40.40.1 255.255.255.0`
	`service cgn cgn1 service-type map-e`
	`!`
	`RP/0/RP0/CPU0:router(config)# interface ServiceApp2`
	`ipv6 address 1001::101/32`
	`service cgn cgn1 service-type map-e`
	`!`
		Configures the application SVI to 1, and enters interface configuration mode.

Step 3	**service cgn** *instance -name* **service-type map-e** **Example:** `RP/0/RP0/CPU0:router(config-if)# service cgn cgn1 service-type map-e map1`	Configures the application SVI to 1, and enters interface configuration mode.
Step 4	**end**or **commit** **Example:**	Saves configuration changes.

	Command or Action	Purpose

	`RP/0/RP0/CPU0:router(config-cgn-map_e)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# commit`
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring a MAP-E Instance

Perform this task to configure a MAP-E instance.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-e** *instance-name*
4. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

	Command or Action	Purpose

Step 3	**service-type map-e** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-e m1`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Configures the service type keyword definition for CGv6 MAP-E application.

Step 4	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Policy Functions

Perform these tasks to configure the policy functions.

Configuring Address Family

Perform these tasks to configure address family.

Configuring IPv4 Address Family

Perform these tasks configure IPv4 address family for a MAP-E instance.

Configuring IPv4 Interface

Perform this task to configure an IPv4 interface for a MAP-E instance.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*

3. **service-type map-e** *instance-name*
4. **address-family ipv4 interface ServiceApp** *number*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**service cgn** *instance-name* **Example:** `RP/0/RP0/CPU0:router(config-if)# service cgn cgn1`	Configures the instance named cgn1 for the CGv6 application, and enters CGv6 configuration mode.

Step 3	**service-type map-e** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-e m1`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)`
		Defines the service type keyword definition for the CGv6 MAP-E application.

Step 4	**address-family ipv4 interface ServiceApp** *number*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# address-family ipv4 interface serviceApp 66`
	`RP/0/RP0/CPU0:router(config-cgn-map_e-afi)#`
		Configures the IPv4 interface to divert IPv4 map-e traffic.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e-afi)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-map_e-afi)#commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring TCP Maximum Segment Size

Perform this task to configure the Maximum Segment Size (MSS) for TCP.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-e** *instance-name*
4. **address-family ipv4 tcp mss** *value*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-e** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-e m1`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Defines the service type keyword definition for the CGv6 MAP-E application.

Step 4	**address-family ipv4 tcp mss** *value*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# address-family ipv4 tcp mss 300`
	`RP/0/RP0/CPU0:router(config-cgn-map_e-afi)#`
		Configures the MSS to be used, in bytes. The range is from 28 to 1500.

Step 5	**end**or **commit** **Example:**	Saves configuration changes. • When you issue the end command, the system prompts you to commit changes:

	Command or Action	Purpose

	`RP/0/RP0/CPU0:router(config-cgn-map_e-afi)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-map_e-afi)# commit`
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring IPv6 Address Family

Perform these tasks configure an IPv6 address family.

Configuring IPv6 Interface

Perform this task to configure an IPv6 interface.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-e** *instance-name*
4. **address-family ipv6 interface ServiceApp** *number*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**service cgn** *instance-name* **Example:**	Configures the instance for the CGv6 application and enters CGv6 configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1` `RP/0/RP0/CPU0:router(config-cgn)#`	

Step 3	**service-type map-e** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-e m1`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Defines the service type keyword definition for the CGv6 MAP-E application.

Step 4	**address-family ipv6 interface ServiceApp** *number*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# address-family ipv6 interface serviceApp 66`
	`RP/0/RP0/CPU0:router(config-cgn-map_e-afi)#`
		Configures the IPv6 interface to divert IPv6 map-e traffic.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e-afi)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-map_e-afi)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring TCP Maximum Segment Size

Perform this task to configure the Maximum Segment Size (MSS) to be used for TCP.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-e** *instance-name*
4. **address-family ipv6 tcp mss** *number*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-e** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-e m1`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Defines the service type keyword definition for the CGv6 MAP-E application.

Step 4	**address-family ipv6 tcp mss** *number*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# address-family ipv6 tcp mss 300`
	`RP/0/RP0/CPU0:router(config-cgn-map_e-afi)#`
		Configures the MSS to be used, in bytes. The range is from 28 to 1500.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e-afi)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-map_e-afi)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring AFTR Endpoint Address

Perform this task to configure the Address Family Transition Router (AFTR) endpoint address.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-e** *instance-name*
4. **aftr-endpoint-address** *IPv6 address*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-e** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-e m1`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Defines the service type keyword definition for the CGv6 MAP-E application.

Step 4	**aftr-endpoint-address** *IPv6 address*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# aftr-endpoint-address 2001:db8::32`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Configures the AFTR endpoint address.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Contiguous Ports

Perform this task to configure the number of contiguous ports for a MAP-E instance

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-e** *instance-name*
4. **contiguous-ports** *number*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-e** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-e m1`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Defines the service type keyword definition for the CGv6 MAP-E application.

	Command or Action	Purpose

Step 4	**contiguous-ports** *number*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# contiguous-ports 16`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Configures the number of contiguous ports. The range is from 1 to 65536.
		Note
		The value is expressed in powers of 2.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring CPE Domain Parameters

Perform this task to configure Customer Premise Equipment (CPE) domain parameters.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-e** *instance-name*
4. **cpe-domain ipv4 prefix** *ipv4 address/prefix* or **cpe-domain ipv6 prefix** *ipv6 address/prefix*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-e** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-e m1`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Defines the service type keyword definition for the CGv6 MAP-E application.

Step 4	**cpe-domain ipv4 prefix** *ipv4 address/prefix* or
	cpe-domain ipv6 prefix *ipv6 address/prefix*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# cpe-domain ipv4 prefix 10.2.2.24/2`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# cpe-domain ipv6 prefix 2001:da8:a464::/48`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Configures the IPv4 or IPv6 prefixes of the CPE domain.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Path MTU of the Tunnel

Perform this task to configure the path Maximum Transmission Unit (MTU) of the tunnel.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-e** *instance-name*
4. **path-mtu** *value*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-e** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-e m1`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Defines the service type keyword definition for the CGv6 MAP-E application.

Step 4	**path-mtu** *value*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# path-mtu 1300`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Configures the path MTU of the tunnel. The range is from 1280 to 9216.

Step 5	**end**or **commit** **Example:**	Saves configuration changes.

	Command or Action	Purpose

	`RP/0/RP0/CPU0:router(config-cgn-map_e)# end`
	or
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# commit`
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Port Sharing Ratio

Perform this task to configure the sharing ratio of the port.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-e** *instance-name*
4. **sharing-ratio** *number*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**service cgn** *instance-name* **Example:**	Configures the instance for the CGv6 application and enters CGv6 configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1` `RP/0/RP0/CPU0:router(config-cgn)#`	

Step 3	**service-type map-e** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-e m1`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Defines the service type keyword definition for the CGv6 MAP-E application.

Step 4	**sharing-ratio** *number*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# sharing-ratio 64`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)#`
		Configures the port sharing ratio. The range is from 1 to 32768.
		Note
		The value is expressed in powers of 2.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-map_e)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

MAP-E on VSM Configuration: Example

This example shows a sample MAP-E configuration on VSM:

```
interface ServiceApp1
ipv4 address 30.30.30.1 255.255.255.0
service cgn cgn1 service-type map-e m1
!
interface ServiceApp2
ipv4 address 19.1.1.1 255.255.255.252
ipv6 address 2001:101::/32
```

```
service cgn cgn1 service-type map-e m1
!
interface ServiceInfra1
ipv4 address 200.1.1.1 255.255.255.0
service-location 0/0/CPU0
!
router static
address-family ipv4 unicast
202.38.102.0/24 ServiceApp1 30.30.30.2
!
address-family ipv6 unicast
2001:da8:a464:ffff::/64 ServiceApp2 2001:101::2
!
service cgn cgn1
service-location preferred-active 0/0/CPU0
service-type map-e m1
cpe-domain ipv6 prefix 2001:da8:a464::/48
cpe-domain ipv4 prefix 202.38.102.0/24
aftr-endpoint-address 2001:da8:a464:ffff::/128
sharing-ratio 16
contiguous-ports 32
path-mtu 1300

address-family ipv4
interface ServiceApp1
tcp mss 235

!
address-family ipv6
interface ServiceApp2
tcp mss 1154
!
!
```

Mapping of Address and Port-Translation Mode

Mapping of Address and Port-Translation Mode (MAP-T) is a CGN solution that enables IPv4-only clients to communicate with IPv6-only resources using address and packet translation. MAP-T is also referred to as Dual IVI (dIVI) or Stateless NAT46. This enables a service provider to offer IPv4 services to IPv6 enabled (customer) sites to which it provides customer premise equipment (CPE). This approach utilizes stateless IPv4 to IPv6 translation (that is NAT64) to transit IPv6-enabled network infrastructure. The provider access network can now be on IPv6, while customers use IPv6 and IPv4 services simultaneously. MAP-T keeps the stateful NAT44 on CPE, as usual, to handle IPv4 address exhaustion, in addition to stateless NAT64 on CPE and Border Router.

MAP-T is attractive to those SPs who have deployed, or are planning to deploy IPv6 end-to-end services, and want to manage IPv4 address exhaustion with utmost predictability.

MAP-T is a preferred alternate to DS-Lite in a sevice provider network when there is no tunneling needed.

Note

- MAP-T is offered in stateless mode only.
- If the cumulative sum of EA bit value and cpe-domain ipv6 prefix value is more than 64 for an interface, then the traffic is dropped for that interface. EA bit value is calculated as (32- ipv4 prefix) + X), where 32 is a constant and X is the power value of 2 when the sharing ratio is expressed as 2^x.

  Ensure that the EA bit value is less than or equal to 32 and sharing ratio is less than or equal to 256 to avoid any traffic drops.
- You should not modify or delete the existing MAP-E and MAP-T configuration. If you modify the existing configuration, the changes are not reflected in the PBR policy.

  To update the configuration, delete the existing MAP-E and MAP-T instance and add new instance with the required changes.
- Do not configure MAP-T and MAP-E on the same CGv6 instance simultaneously.

Configuring MAP-T

Perform these tasks to configure MAP-T.

Note MAP-T is supported only on Cisco ASR 9000 Series 400G and 200G Modular Line Cards and Cisco ASR 9000 Series 4-Port and 8-Port 100 Gigabit Ethernet Line Cards.

Configuring the Application Service Virtual Interface

This section lists the guidelines for selecting service application interfaces for MAP-T.

- Pair ServiceApp<n> with ServiceApp<n+1>, where <n> is an odd integer. This is to ensure that the ServiceApp pairs works with a maximum throughput. For example, ServiceApp1 with ServiceApp2 or ServiceApp3 with ServiceApp4.
- Pair ServiceApp<n> with ServiceApp<n+5> or ServiceApp<n+9>, and so on, where <n> is an odd integer. For example, ServiceApp1 with ServiceApp6, ServiceApp1 with ServiceApp10, ServiceApp3 with ServiceApp8, or ServiceApp3 with ServiceApp12.
- Pair ServiceApp<n> with ServiceApp<n+4>, where <n> is an integer (odd or even integer). For example, ServiceApp1 with ServiceApp5, or ServiceApp2 with ServiceApp6.

Warning Although ServiceApp pairs work, the aggregate throughput for Inside-to-Outside and Outside-to-Inside traffic for the ServiceApp pair is halved.

Caution Do not pair ServiceApp<n> with ServiceApp<n+1>, where <n> is an even integer. When used, Outside-to-Inside traffic is dropped because traffic flows in the incorrect dispatcher and core.

Perform this task to configure the application service virtual interface (SVI) to forward data traffic.

SUMMARY STEPS

1. **configure**
2. **interface ServiceApp** *value*
3. **service cgn** *instance-name* **service-type map-t**
4. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**interface ServiceApp** *value*
	Example:
	`RP/0/RP0/CPU0:router(config)# interface ServiceApp 1`
	`RP/0/RP0/CPU0:router(config-if)#`
		Configures the application SVI to 1, and enters interface configuration mode.

Step 3	**service cgn** *instance-name* **service-type map-t** **Example:** `RP/0/RP0/CPU0:router(config-if)# service cgn cgn1 service-type map-t map1`	Configures the application SVI to 1, and enters interface configuration mode.

Step 4	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring a MAP-T Instance

Perform this task to configure a MAP-T instance.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-t** *instance-name*
4. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance named cgn1 for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-t** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-t map-t-inst`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#`
		Configures the service type keyword definition for CGv6 MAP-T application

Step 4	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		Uncommitted changes found, commit them before exiting (yes/no/cancel)?

	Command or Action	Purpose
		[cancel]: • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Policy Functions

Perform these tasks to configure the policy functions.

Configuring Address Family

Perform these tasks to configure address family.

Configuring IPv4 Address Family

Perform these tasks configure IPv4 address family for a MAP-T instance.

Configuring an IPv4 Interface

Perform this task to configure an IPv4 interface for a MAP-T instance.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-t** *instance-name*
4. **address-family ipv4 interface ServiceApp** *number*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-t** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-t map-t-inst`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#`
		Configures the service type keyword definition for CGv6 MAP-T application.

Step 4	**address-family ipv4 interface ServiceApp** *number*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#address-family ipv4 interface serviceApp 66`
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)`
		Configures the IPv4 interface to divert IPv4 MAP-T traffic.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring IPv4 TCP Maximum Segment Size (MSS)

Perform this task to configure the MSS for TCP in bytes.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-t** *instance-name*
4. **address-family ipv4 tcp mss** *value*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-t** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-t map-t-inst`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#`
		Configures the service type keyword definition for CGv6 MAP-T application.

Step 4	**address-family ipv4 tcp mss** *value*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#address-family ipv4 tcp mss 66`
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)`
		Configures the MSS for TCP in bytes.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring IPv4 Type of Service (ToS)

Perform this task to configure the configured ToS value to be used when translating a packet from IPv6 to IPv4.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-t** *instance-name*
4. **address-family ipv4 tos** *value*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-t** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-t map-t-inst`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#`
		Configures the service type keyword definition for CGv6 MAP-T application.

	Command or Action	Purpose

Step 4	**address-family ipv4 tos** *value*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#address-family ipv4 tos 66`
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)`
		Configures the TOS value.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring IPv6 Address Family

Perform these tasks configure an IPv6 address family.

Configuring IPv6 Do not Fragment (DF) Override

Perform this task to enable DF override configuration.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-t** *instance-name*
4. **address-family ipv6 df-override**
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-t** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-t map-t-inst`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)`
		Configures the service type keyword definition for CGv6 MAP-T application.

Step 4	**address-family ipv6 df-override**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#address-family ipv6 df-override`
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)`
		Configures the DF-Override.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring an IPv6 Interface

Perform this task to configure an IPv6 interface for a stateful NAT64 instance.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat64 stateful** *instance-name*
4. **address-family ipv6 interface ServiceApp** *number*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type nat64 stateful** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type nat64 stateful nat64-inst`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)`
		Configures the service type keyword definition for CGv6 Stateful NAT64 application.

Step 4	**address-family ipv6 interface ServiceApp** *number*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)#address-family ipv4 interface ServiceApp 66`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stful-afi)`
		Configures the IPv6 interface to divert IPv6 nat64 traffic.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stful-afi)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stful-afi)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring IPv6 TCP Maximum Segment Size (MSS)

Perform this task to configure the MSS for TCP in bytes.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-t** *instance-name*
4. **address-family ipv6 tcp mss** *value*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-t** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-t map-t-inst`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)`
		Configures the service type keyword definition for CGv6 MAP-T application.

Step 4	**address-family ipv6 tcp mss** *value* **Example:**	Configures the MSS for TCP in bytes.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#address-family ipv6 tcp mss 66` `RP/0/RP0/CPU0:router(config-cgn-mapt-afi)`	

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-mapt-afi)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring IPv6 Traffic-Class

Perform this task to configure a traffic-class.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type nat64 stateful** *instance-name*
4. **address-family ipv6 traffic-class** *value*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type nat64 stateful** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type nat64 stateful nat64-inst`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)`
		Configures the service type keyword definition for CGv6 Stateful NAT64 application.

Step 4	**address-family ipv6 traffic-class** *value*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stateful)#address-family ipv6 traffic-class 66`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stful-afi)`
		Configures the traffic class to be set.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stful-afi)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-nat64-stful-afi)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Contiguous Ports

Perform this task to configure contiguous ports.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-t** *instance-name*

4. **contiguous-ports** *number*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-t** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-t map-t-inst`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)`
		Configures the service type keyword definition for CGv6 MAP-T application.

Step 4	**contiguous-ports** *number*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#contiguous-ports 14`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)`
		Configures the number of ports and the value is expressed in powers of 2. The range is from 1 to 65536.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Customer Premise Equipment Domain Parameters

Perform this task to configure Customer Premise Equipment (CPE) domain parameters.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-t** *instance-name*
4. **cpe-domain ipv4 prefix** *ipv4 address/prefix* or **cpe-domain ipv6 prefix** *ipv6 address/prefix*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-t** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-t map-t-inst`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)`
		Configures the service type keyword definition for CGv6 MAP-T application.

Step 4	**cpe-domain ipv4 prefix** *ipv4 address/prefix* or **cpe-domain ipv6 prefix** *ipv6 address/prefix*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#acpe-domain ipv4 prefix 10.2.2.24/2`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#acpe-domain ipv6 prefix 10:2::2/24`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)`
		Configures the cpe domain parameters.

	Command or Action	Purpose

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring External Domain Parameters

Perform this task to configure external domain parameters.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-t** *instance-name*
4. **external-domain ipv6 prefix** *ipv6 address/prefix*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**service cgn** *instance-name* **Example:**	Configures the instance for the CGv6 application and enters CGv6 configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1` `RP/0/RP0/CPU0:router(config-cgn)#`	

Step 3	**service-type map-t** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-t map-t-inst`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)`
		Configures the service type keyword definition for CGv6 MAP-T application.

Step 4	**external-domain ipv6 prefix** *ipv6 address/prefix*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#external-domain ipv6 prefix 10:2::2/24`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)`
		Configures the external domain parameters.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Port Sharing Ratio

Perform this task to configure port sharing ratio.

SUMMARY STEPS

1. **configure**
2. **service cgn** *instance-name*
3. **service-type map-t** *instance-name*
4. **sharing-ratio** *number*
5. **end**or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgn** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgn cgn1`
	`RP/0/RP0/CPU0:router(config-cgn)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-type map-t** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn)# service-type map-t map-t-inst`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)`
		Configures the service type keyword definition for CGv6 MAP-T application.

Step 4	**sharing-ratio** *number*
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)#sharing-ratio 14`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)`
		Configures the port sharing ratio and the value is expressed in powers of 2. The range is from 1 to 32768.

Step 5	**end**or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cgn-mapt)# end`
	`or`
	`RP/0/RP0/CPU0:router(config-cgn-mapt)# commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

MAP-T Configuration on VSM: Example

```
interface ServiceApp4
 ipv4 address 30.30.30.1 255.255.255.0
 service cgn test service-type map-t
!
interface ServiceApp6
 ipv4 address 19.1.1.1 255.255.255.252
 ipv6 address 2001:101::/32
 service cgn test service-type map-t
!
interface ServiceInfra1
 ipv4 address 200.1.1.1 255.255.255.0
 service-location 0/0/CPU0
!
router static
 address-family ipv4 unicast
202.38.102.0/24 ServiceApp4 30.30.30.2
  !
 address-family ipv6 unicast
  2001:da8:a464:ffff::/64 ServiceApp6 2001:101::2
!
service cgn test
 service-location preferred-active 0/0/CPU0
service-type map-t xlat1
  cpe-domain ipv6 prefix 2001:da8:a464::/48
  cpe-domain ipv4 prefix  202.38.102.0/24
  external-domain ipv6 prefix 2001:da8:a464:ffff::/64
  sharing-ratio 64
  contiguous-ports 128

       address-family ipv4
   interface ServiceApp4
 tcp mss 235
 tos 100
  !
      address-family ipv6
   interface ServiceApp6
  tcp mss 1154
 traffic-class 100
 df-override
;
 !
!
```

Configuration Examples for Implementing the Carrier Grade NAT

This section provides the following configuration examples for CGN:

Note Do not configure multiple outside address-pools to be mapped to a single inside-vrf. If you have multiple outside address-pools to be mapped, then create multiple inside-vrfs and map each outside address-pool to a single inside-vrf inside the NAT44 configuration.

Configuring a Different Inside VRF Map to a Different Outside VRF: Example

This example shows how to configure a different inside VRF map to a different outside VRF and different outside address pools:

```
service cgn cgn1
inside-vrf insidevrf1
map outside-vrf outsidevrf1 address-pool 100.1.1.0/24
!
!
inside-vrf insidevrf2
map outside-vrf outsidevrf2 address-pool 100.1.2.0/24
!
service-location preferred-active 0/2/cpu0 preferred-standby 0/3/cpu0
!
interface ServiceApp 1
vrf insidevrf1
ipv4 address 210.1.1.1 255.255.255.0
service cgn cgn1
!
router static
vrf insidevrf1
0.0.0.0/0 serviceapp 1
!
!
interface ServiceApp 2
vrf insidevrf2
ipv4 address 211.1.1.1 255.255.255.0
service cgn cgn1
service-type nat44 nat1
!
router static
vrf insidevrf2
0.0.0.0/0 serviceapp 2
!
!
interface ServiceApp 3
vrf outsidevrf1
ipv4 address 1.1.1.1 255.255.255.0
service cgn cgn1
service-type nat44 nat1
!
router static
vrf outsidevrf1
100.1.1.0/24 serviceapp 3
!
!
interface ServiceApp 4
vrf outsidevrf2
ipv4 address 2.2.2.1 255.255.255.0
service cgn cgn1
service-type nat44 nat1
!
router static
vrf outsidevrf2
100.1.2.0/24 serviceapp 4
!
```

Configuring a Different Inside VRF Map to a Same Outside VRF: Example

This example shows how to configure a different inside VRF map to the same outside VRF but with different outside address pools:

```
service cgn cgn1
inside-vrf insidevrf1
map outside-vrf outsidevrf1 address-pool 100.1.1.0/24
!
inside-vrf insidevrf2
map outside-vrf outsidevrf1 address-pool 200.1.1.0/24
!
!
service-location preferred-active 0/2/cpu0 preferred-standby 0/3/cpu0
!
interface ServiceApp 1
vrf insidevrf1
ipv4 address 1.1.1.1 255.255.255.0
service cgn cgn1
!
router static
vrf insidevrf1
0.0.0.0/0 serviceapp 1
!
!
interface ServiceApp 2
vrf insidevrf2
ipv4 address 2.1.1.1 255.255.255.0
service cgn cgn1
!
router static
vrf insidevrf2
0.0.0.0/0 serviceapp 2
!
!
interface ServiceApp 3
vrf outsidevrf1
ipv4 address 100.1.1.1 255.255.255.0
service cgn cgn1
!
router static
vrf outsidevrf1
100.1.1.0/24 serviceapp 3
200.1.1.0/24 serviceapp 3
!
```

NAT44 Configuration: Example

This example shows a NAT44 sample configuration:

```
IPv4: 40.22.22.22/16
!
interface Loopback40
 description IPv4 Host for NAT44
 ipv4 address 40.22.22.22 255.255.0.0
!
interface Loopback41
 description IPv4 Host for NAT44
 ipv4 address 41.22.22.22 255.255.0.0
!
interface GigabitEthernet0/3/0/0.1
 description Connected to P2_CRS-8 GE 0/6/5/0.1
```

```
 ipv4 address 10.222.5.22 255.255.255.0
 encapsulation dot1q 1
!
router static
 address-family ipv4 unicast
  180.1.0.0/16 10.222.5.2
  181.1.0.0/16 10.222.5.2
!
!
Hardware Configuration for CSGE:
!
vrf InsideCustomer1
 address-family ipv4 unicast
 !
!
vrf OutsideCustomer1
 address-family ipv4 unicast
 !
!
hw-module service cgn location 0/3/CPU0
!
service-plim-ha location 0/3/CPU0 datapath-test
service-plim-ha location 0/3/CPU0 core-to-core-test
service-plim-ha location 0/3/CPU0 pci-test
service-plim-ha location 0/3/CPU0 coredump-extraction
!
!
interface GigabitEthernet0/6/5/0.1
 vrf InsideCustomer1
 ipv4 address 10.222.5.2 255.255.255.0
 encapsulation dot1q 1
!
interface GigabitEthernet0/6/5/1.1
 vrf OutsideCustomer1
 ipv4 address 10.12.13.2 255.255.255.0
 encapsulation dot1q 1
!
interface ServiceApp1
 vrf InsideCustomer1
 ipv4 address 1.1.1.1 255.255.255.252
 service cgn cgn1 service-type nat44
!
interface ServiceApp2
 vrf OutsideCustomer1
 ipv4 address 2.1.1.1 255.255.255.252
 service cgn cgn1 service-type nat44
!
interface ServiceInfra1
 ipv4 address 75.75.75.75 255.255.255.0
 service-location 0/3/CPU0
!
!
router static
 !
vrf InsideCustomer1
  address-family ipv4 unicast
   0.0.0.0/0 ServiceApp1
   40.22.0.0/16 10.222.5.22
   41.22.0.0/16 10.222.5.22
   181.1.0.0/16 vrf OutsideCustomer1 GigabitEthernet0/6/5/1.1 10.12.13.1
  !
 !
 vrf OutsideCustomer1
  address-family ipv4 unicast
```

```
   40.22.0.0/16 vrf InsideCustomer1 GigabitEthernet0/6/5/0.1 10.222.5.22
   41.22.0.0/16 vrf InsideCustomer1 GigabitEthernet0/6/5/0.1 10.222.5.22
   100.0.0.0/24 ServiceApp2
   180.1.0.0/16 10.12.13.1
   181.1.0.0/16 10.12.13.1
  !
 !
!
CGSE Configuration:
service cgn cgn1
 service-location preferred-active 0/3/CPU0
 service-type nat44 nat44
  portlimit 200
  alg ActiveFTP
  inside-vrf InsideCustomer1
   map outside-vrf OutsideCustomer1 address-pool 100.0.0.0/24
   protocol tcp
    static-forward inside
     address 41.22.22.22 port 80
    !
   !
   protocol icmp
    static-forward inside
     address 41.22.22.22 port 80
    !
   !
   external-logging netflow version 9
    server
     address 172.29.52.68 port 2055
     refresh-rate 600
     timeout 100 !
   !
  !
 !
!
IPv4: 180.1.1.1/16
!
interface Loopback180
 description IPv4 Host for NAT44
 ipv4 address 180.1.1.1 255.255.0.0
!
interface Loopback181
 description IPv4 Host for NAT44
 ipv4 address 181.1.1.1 255.255.0.0
!
interface GigabitEthernet0/6/5/1.1
 ipv4 address 10.12.13.1 255.255.255.0
 encapsulation dot1q 1
!
router static
 address-family ipv4 unicast
  40.22.0.0/16 10.12.13.2
  41.22.0.0/16 10.12.13.2
  100.0.0.0/24 10.12.13.2 !
!
```

NAT64 Configuration: Example

This example shows a NAT64 sample configuration:

```
! Defines the Location for CGN Translation
! ---------------------------------
hw-module service cgn location 0/1/cpu0
```

```
! Defines the Service Infra Interface
! -----------------------------
interface ServiceInfra 1

  ! Defines the IP address and netmask
  ipv4 address 3.1.1.2 255.255.255.252

! Defines IPv4 to IPv6 Direction Service Application Interface
! --------------------------------------------
interface ServiceApp 1

  ! Assigns the IPv4 Address and Netmask
  ipv4 address 211.1.1.1 255.255.255.0

  ! Indicates the CGN Instance and Service-type
  service cgn cgn1 service-type nat64 stateful

! Defines IPv6 to IPv4 direction Service Application interface
! --------------------------------------------
interface ServiceApp 2

  ! Assigns IPv6 address and netmask
  ipv6 address 5005::5555/96

  ! Indicates the CGN instance and Service-type
  service cgn cgn1 service-type nat64 stateful

! Define the CGN Instance
! --------------------
service cgn cgn1

  ! Defines the Location
  service-location preferred-active 0/1/CPU0

  ! Defines the Service-type and Name
  service-type nat64 stateful nat64_1

    ! Enables RTSP ALG
    alg RTSP

    ! Defines the Port Limit
    portlimit 65535

    ! Defines IPv6 Prefix
    ipv6-prefix 3301:db8:1::/96

    ! Defines the IPv4 Address Pool
    ipv4 address-pool 52.1.64.0/22

    ! Defines the Dynamic Port Range
    dynamic-port-range start 1

    ! Defines the IPv4 Address Family
    address-family ipv4
        interface ServiceApp1

    ! Defines IPv6 Address Family
    address-family ipv6
      interface ServiceApp2

    ! Defines the UDP/TCP/ICMP Protocol
    protocol udp
      timeout 65535
```

```
    protocol tcp
      session initial timeout 65535
      session active timeout 65535

    protocol icmp
      timeout 65535

    ! Defines the Netflow and Associated Server.
    external-logging netflow version 9
      server
        address 10.64.81.232 port 44444

! Defines the Static Route for IPv4 to IPv6 direction
address-family ipv4 unicast

  ! This route is the same as the IPv4 CPE Domain Prefix
  52.1.64.0/22 ServiceApp1

! Defines Static Route for IPv6 to IPv4 direction
address-family ipv6 unicast

  ! This route is same as the IPv6 External Domain Prefix
  3301:db8:1::/96 ServiceApp2
```

Bulk Port Allocation and Syslog Configuration: Example

```
service cgn cgn2
 service-type nat44 natA
  inside-vrf broadband
   map address-pool 100.1.2.0/24
   external-logging syslog
    server
     address 20.1.1.2 port 514
    !
   !
   bulk-port-alloc size 64
  !
 !
```

Predefined NAT Configuration: Example

This example shows how to configure the predefined NAT for NAT44:

```
service cgn cgn1
service-location preferred-active 0/2/CPU0
service-type nat44 nat1
 inside-vrf red
   map outside-vrf blue address-pool 100.0.0.0/24
   nat-mode
    predefined private-pool 103.1.106.0/24
    predefined private-pool 103.1.107.0/26
    predefined private-pool 103.1.107.128/26
    predefined private-pool 103.1.108.0/23
    predefined private-pool 103.1.112.0/22
    predefined private-pool 103.1.116.0/24
    predefined private-pool 103.1.117.64/26
    predefined private-pool 103.1.117.192/26
```

PPTP ALG Configuration: Example

NAT44 Instance

```
service cgn cgn1
 service-location preferred-active 0/1/CPU0
 service-type nat44 inst1
  alg pptpAlg
```

•

DBL Configuration: Example

NAT44 Instance

```
service cgn cgn1
 service-type nat44 nat1
  inside-vrf ivrf
   external-logging netflow version 9
    server
     address x.x.x.x port x
      session-logging
```

Configuring TCP Sequence-Check: Example

This example shows how to configure sequence check for TCP sessions.

```
configure
service cgn cgn1
service-type nat44 nat1
inside-vrf vrf1
firewall protocol tcp
sequence-check
```

Configuring Address and Port-Dependent Filtering: Example

This example shows how to configure address and port-dependent filtering.

```
configure
service cgn cgn1
service-type nat44 nat1
inside-vrf vrf1
filter-policy ignore-port
```

NAT0 Mode Configuration: Example

This example shows how to configure the NAT0 mode:

```
service cgn cgn1
 service-location preferred-active 0/2/CPU0
  service-type nat44 nat1
    inside-vrf Inside_1
      map outside-vrf ovrf outsideServiceApp ServiceApp2 address-pool 0.0.0.0/0
```

```
      nat-mode no-nat
```

Configuration of Multiple NetFlow Servers: Example

```
service cgn cgn1
 service-location preferred-active 0/1/CPU0
 service-type nat44 nat1
  inside-vrf ivrf
   map outside-vrf ovrf outsideServiceApp ServiceApp2 address-pool 100.1.1.0/24
    external-logging netflow version 9
    server
     address 111.1.1.1 port 6000
      path-mtu 1200
      refresh-rate 600
      timeout 1000
      session-logging
     !
     address 111.1.1.1 port 9000
      path-mtu 1100
      refresh-rate 500
      timeout 1000
      session-logging
     !
     address 122.1.1.1 port 9000
      path-mtu 1200
      refresh-rate 500
      timeout 1100
      session-logging
     !
    !
   !
```

Configuration of Multiple Syslog Servers: Servers

```
service cgn cgn1
 service-location preferred-active 0/1/CPU0
 service-type nat44 nat1
  inside-vrf ivrf
   map outside-vrf ovrf outsideServiceApp ServiceApp2 address-pool 100.1.1.0/24
   external-logging syslog
    server
     address 211.1.1.1 port 6000
      path-mtu 1200
      session-logging
     !
     address 211.1.1.1 port 9000
      path-mtu 1200
      session-logging
     !
     address 212.1.1.1 port 6000
      path-mtu 1200
      session-logging
     !
    !
   !
```

CGN Sequential Allocation Algorithm

In classic NAT, the process of mapping a private IP to a public IP or a private port to an outside port is random. Therefore, it becomes difficult to track the subscribers using an IP and a port at a given time. Predefined NAT avoids this random process by mapping a private IP address to a range of ports associated with the corresponding public IP address. This is done through an algorithm that helps the user to recognize a private IP address without having to refer to the massive CGN logs. The address and port translation is done in accordance with the algorithm.

CGN sequential algorithm is based on RFC 7422 support for CGNAT44 deployment with A9K-VSM-500.

With CGN sequential algorithm, you can perform the following:

- Configure **dynamic port start range**, **port block allocation size**, **inside IPv4 pool**, **outside IPv4 pool**.
- Specify inside and outside pools by a start address and end address, instead of IPv4 prefix.

Limitations

- Public Address Pool and Private address Pool distribution across cores is now dependent on port limit and dynamic port start values
- Default value of Dynamic port start is 1024 and Port Limit default value is 2048
- Port Limit minimum value is 256 and Maximum Value is 16384 for Sequential NAT
- Only One Sequential Pool can be configured per inside VRF. You have to mention ServiceApp number with Outside VRF in Public Pool configuration
- Show Output Translation command in sequential NAT 44 displays entries as per cores instead of blocks



Note All the limitations of Predefined DET NAT are applicable to Sequential Predefined NAT.

Limitations of Predefined DET NAT

- The Bulk Port Allocation configuration is not available in the predefined mode. If you try to configure, Bulk Port Allocation on an inside VRF that has the predefined mode enabled, the configuration is rejected during verification.
- The port-preservation option is not available in the predefined mode.
- The global port limit parameter is not available for the predefined mode. Even though you will be allowed to configure the global port limit, the inside VRF, which has predefined mode enabled, ignores that port limit and uses the port limit configured by the algorithm.
- If you turn the predefined mode on or off for an inside VRF during the active translations, all the translations on that VRF are deleted.
- Ensure that you configure NetFlow or syslog only if it is very much required.

- Any configuration change that results in changes in mapping deletes the existing translations. Therefore, ensure that you record such configuration changes. You might need this information to trace the port usage by a subscriber.
- Ensure uniform port allocation uniform for all subscribers.

Configuring Sequential Predefined NAT

1. configure terminal
2. service cgn instance-name
3. service-location preferred-active 0/x/CPU0 (‘x’ location of VSM card)
4. service-type nat44 nat1
5. dynamic-port-range start 2048
6. inside-vrf vrf-name
7. nat-mode
8. predefined seq-private-pool start-address end-address
9. map outside-vrf vrf-name seq-address-pool start-address end-address
10. portlimit 2048
11. commit

Sequential Predefined NAT Configuration: Example

```
service cgn cgn123
service-location preferred-active 0/3/CPU0
service-type nat44 nat1
dynamic-port-range start 2048
   inside-vrf red
   nat-mode
   predefined seq-private-pool 12.0.0.0 12.0.0.255
   !
   map outside-vrf blue outsideServiceApp ServiceApp2 seq-address-pool 100.0.0.0 100.0.0.255

   portlimit 2048
  !
```

Verification

```
RP/0/RSP0/CPU0:router#sh run service cgn cgn123
service cgn cgn123
 service-location preferred-active 0/3/CPU0
 service-type nat44 nat1
  dynamic-port-range start 2048
  inside-vrf red
   nat-mode
    predefined seq-private-pool 12.0.0.0 12.0.0.255
   !
   map outside-vrf blue seq-address-pool 100.0.0.0 100.0.0.255
   portlimit 2048
  !
```

```
 !

RP/0/RSP0/CPU0:router#sh cgn nat44 nat1 statistics
Statistics summary of NAT44 instance: 'nat1'
Number of active translations: 10496
Number of sessions: 0
Translations create rate: 0
Translations delete rate: 0
Inside to outside forward rate: 4078
Outside to inside forward rate: 4396
Inside to outside drops port limit exceeded: 0
Inside to outside drops system limit reached: 0
Inside to outside drops resource depletion: 0
No translation entry drops: 0
PPTP active tunnels: 0
PPTP active channels: 0
PPTP ctrl message drops: 0
Number of subscribers: 255
Drops due to session db limit exceeded: 0
Drops due to source ip not configured: 0

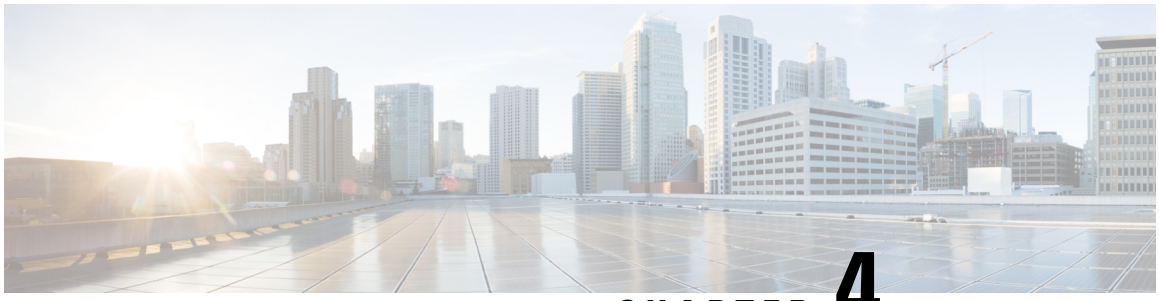
Pool address totally free: 0
Pool address used: 256
Pool address usage:
------------------------------------------------
  External Address       Ports Used
------------------------------------------------
  100.0.0.0              41
  100.0.0.0              41
  100.0.0.1              41
  100.0.0.2              41
  100.0.0.3              41
  100.0.0.3              41
  100.0.0.4              41
  100.0.0.5              41
……..
RP/0/RSP0/CPU0:router#show cgn nat44 nat1 mapping inside-address inside-vrf red start-addr
 12.0.0.0 end-addr 12.0.0.255
Mapping details for address pool inside a Vrf
-------------------------------------------------------------------------------------------------
NAT44 instance    : nat1
VRF               : red
-------------------------------------------------------------------------------------------------
Inside Ip          Outside IP       Type                Port Range         Ports Used
Address             Address
-------------------------------------------------------------------------------------------------
12.0.0.0         100.0.0.0            Predefined       2048-4095              41
12.0.0.1         100.0.0.0            Predefined       4096-6143              41
12.0.0.2         100.0.0.0            Predefined       6144-8191              41
12.0.0.3         100.0.0.0            Predefined       8192-10239             41
12.0.0.4         100.0.0.0            Predefined      10240-12287             41
12.0.0.5         100.0.0.0            Predefined      12288-14335             41
12.0.0.6         100.0.0.0            Predefined      14336-16383             41
12.0.0.7         100.0.0.0            Predefined      16384-18431             41
12.0.0.8         100.0.0.0            Predefined      18432-20479             41
12.0.0.9         100.0.0.0            Predefined      20480-22527             41
12.0.0.10        100.0.0.0            Predefined      22528-24575             41
12.0.0.11        100.0.0.0            Predefined      24576-26623             41
12.0.0.12        100.0.0.0            Predefined      26624-28671             41
12.0.0.13        100.0.0.0            Predefined      28672-30719             41
12.0.0.14        100.0.0.0            Predefined      30720-32767             41
……..
```

When **o2i-vrf-override** keyword is used for SEQ-NAT44, **ipv4 forwarding-enable** has to be configured on 12 TenGige (VNIC) interfaces on the VSM location that will receive reverse translated traffic. Refer to the config below:

```
interface TenGigE0/1/1/0
description virtual-service interface
mtu 9126
ipv4 forwarding-enable
!
```



Note The configuration is same for all the other 12 VNIC interfaces (interface TenGigE0/1/1/1, interface TenGigE0/1/1/2, and so on, to interface TenGigE0/1/1/12).



CHAPTER 4

Carrier Grade IPv6 without Service Modules

This module describes how to implement the Carrier Grade IPv6 (CGv6) without Services Modules.

- MAP-E without service modules, on page 129
- Configuring MAP-E without service modules , on page 130
- MAP-T without Service Cards, on page 137
- Overview of MAP-T Logging, on page 158

MAP-E without service modules

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
MAP-E support on 5th Generation ASR 9000 Series Line Cards	Release 7.4.1	MAP-E has been supported on Cisco ASR 9000 Series and Cisco ASR 9900 series 4th generation Ethernet line cards. In this release, Map-E support is extended to the following Cisco ASR9000 Series and Cisco ASR 9900 Series 5th generation hardware: • A9K-20HG-FLEX-SE/TR • A9K-8HG-FLEX-SE/TR • A99-32X100GE-X-SE/TR • A99-10X400GE-X-SE/TR • A9K-4HG-FLEX-SE/TR • ASR-9903 fixed-port router and Port Expansion Cards (PECs) • ASR-9902 fixed-port router

This feature configures Mapping of Address and Portal-Encapsulation Mode (MAP-E) CGN solution without service cards (ISM or VSM). The CGN application directly interacts with the line cards to configure MAP-E.



Note The MAP-E CGN solution without service cards (VSM/ISM) is supported on Cisco IOS XR and Cisco IOS XR 64 bit operating system.

Restrictions for Configuring MAP-E without Service Card on Cisco IOS XR 32-bit Operating System

- MAP-E is supported on the 3rd and 4th generation of Cisco ASR 9000 Series Ethernet line cards.
- MAP-E without service cards can be enabled only in a default VRF.
- If MAP-E or MAP-T is enabled on an interface, other policy based routing (PBR) features such as CLI PBR, BGP Flow Spec, One Platform Kit (onePK), OpenFlow, ingress ACLs and BNG are not supported. This is because only one PBR policy is allowed on the interface per direction.
- In a router, only one mode of either inline-service with service card or inline-service without service card is supported.

Restrictions for Configuring MAP-E without Service Card on Cisco IOS XR 64-bit Operating System

- From Cisco IOS XR Release 7.0.1 onwards, Cisco ASR 9000 Series 4th Generation Ethernet line cards support MAP-E.
- From Cisco IOS XR Release 7.4.1 onwards, Cisco ASR 9000 Series 5th Generation Ethernet line cards support MAP-E.

Types of exception packets handled only by Service Module:

- IPv6 extension headers.
- V4/V6 fragmented packets.
- ICMP messages (excluding ICMP echo message and reply packets, which are processed by the inline interface for MAP-E)
- TCP Maximum Segment Size and Path MTU checks.
- Packets with Loose Source Route (LSR) and Strict Source Route (SSR) IPv4 options

Configuring MAP-E without service modules

This feature allows to configure Mapping of Address and Portal-Encapsulation Mode (MAP-E) CGN solution without service cards (ISM or VSM). The CGN application directly interacts with the line cards to configure MAP-E.

Configuring MAP-E instances without service modules

Perform these tasks to configure MAP-E without service modules.

SUMMARY STEPS

1. **configure**
2. **service cgv6** *instance-name*
3. **service-inline***interface-name*
4. **service-type map-e** *instance-name*
5. **end** or **commit**

DETAILED STEPS

Procedure

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

Step 2	**service cgv6** *instance-name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# service Cgv6 cgv6-1`
	`RP/0/RSP0/CPU0:router(config-cgv6)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.
		Note
		The maximum number of CGv6 applications allowed for a MAP-E instance is 6.

Step 3	**service-inline***interface-name* **Example:** `RP/0/RSP0/CPU0:router(config-cgv6)#Service-inline interface TenGigE0/0/0/0`	Configures the service-inline interface.

Step 4	**service-type map-e** *instance-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-cgv6) #service-type map-e map1`
	`RP/0/RSP0/CPU0:router(config-cgV6-map_e)#`
		Configures the service type keyword definition for CGv6 MAP-E application.

Step 5	**end** or **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)# end`
	`or`
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)# Commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

	Command or Action	Purpose
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring cpe domain parameters without service modules

Perform this task to configure cpe domain interface for MAP-E without service modules.

SUMMARY STEPS

1. **configure**
2. **service cgv6** *instance-name*
3. **service-inline** *interface-name*
4. **service-type map-e** *instance-name*
5. **cpe-domain ipv4 prefix** *ipv4 adress/prefix* **cpe-domain ipv6 prefix** *ipv6 adress/prefix*
6. **end** or **commit**

DETAILED STEPS

Procedure

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

Step 2	**service cgv6** *instance-name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# service Cgv6 cgv6-1`
	`RP/0/RSP0/CPU0:router(config-cgv6)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-inline** *interface-name* **Example:** `RP/0/RSP0/CPU0:router(config-cgv6)#Service-inline interface TenGigE0/0/0/0`	Configures the service-inline interface.
Step 4	**service-type map-e** *instance-name* **Example:**	Configures the service type keyword definition for CGv6 MAP-E application.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-cgv6) #service-type map-e map1` `RP/0/RSP0/CPU0:router(config-cgV6-map_e)#`	

Step 5	**cpe-domain ipv4 prefix** *ipv4 adress/prefix* **cpe-domain ipv6 prefix** *ipv6 adress/prefix*
	Example:
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)#cpe-domain ipv4 prefix 120.2.1.0/24`
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)#cpe-domain ipv6 prefix 9020:da8:2::/48`
		Configures the IPv4 or IPv6 prefixes of the CPE domain parameter without service modules.

Step 6	**end** or **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)# end`
	`or`
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)# Commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring port sharing ratio and contiguous port without service modules

Perform this task to configure port sharing ratio and contiguous port.

SUMMARY STEPS

1. **configure**
2. **service cgv6** *instance-name*
3. **service-inline** *interface-name*
4. **service-type map-e** *instance-name*
5. **cpe-domain ipv4 prefix** *ipv4 adress/prefix* **cpe-domain ipv6 prefix** *ipv6 adress/prefix*
6. **sharing-ration 256**
7. **contiguous-port 16**

8. **end** or **commit**

DETAILED STEPS

Procedure

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

Step 2	**service cgv6** *instance-name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# service Cgv6 cgv6-1`
	`RP/0/RSP0/CPU0:router(config-cgv6)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-inline** *interface-name* **Example:** `RP/0/RSP0/CPU0:router(config-cgv6)#Service-inline interface TenGigE0/0/0/0`	Configures the service-inline interface.

Step 4	**service-type map-e** *instance-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-cgv6) #service-type map-e map1`
	`RP/0/RSP0/CPU0:router(config-cgV6-map_e)#`
		Configures the service type keyword definition for CGv6 MAP-E application.

Step 5	**cpe-domain ipv4 prefix** *ipv4 adress/prefix* **cpe-domain ipv6 prefix** *ipv6 adress/prefix*
	Example:
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)#cpe-domain ipv4 prefix 120.2.1.0/24`
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)#cpe-domain ipv6 prefix 9020:da8:2::/48`
		Configures the IPv4 or IPv6 prefixes of the CPE domain parameter without service modules.

Step 6	**sharing-ration 256** **Example:** `RP/0/RSP0/CPU0:router(config-cgv6-map_e)#sharing-ratio 256`	Configures the port sharing ratio. The value for the port sharing ratio is 256.
Step 7	**contiguous-port 16** **Example:** `RP/0/RSP0/CPU0:router(config-cgv6-map_e)#contiguous-ports 16`	Configures the contiguous port. The value for the contiguous port is 16.
Step 8	**end** or **commit**	Saves configuration changes.

	Command or Action	Purpose

	Example:
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)# end`
	or
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)# Commit`
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring BR Endpoint Address without modules

Perform this task to configure the BR Endpoint Address without service modules.

SUMMARY STEPS

1. **configure**
2. **service cgv6** *instance-name*
3. **service-inline** *interface-name*
4. **service-type map-e** *instance-name*
5. **cpe-domain ipv4 prefix** *ipv4 adress/prefix* **cpe-domain ipv6 prefix** *ipv6 adress/prefix*
6. **sharing-ration 256**
7. **contiguous-port 16**
8. **br-endpoint-adress**
9. **end**or **commit**

DETAILED STEPS

Procedure

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

	Command or Action	Purpose

Step 2	**service cgv6** *instance-name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# service Cgv6 cgv6-1`
	`RP/0/RSP0/CPU0:router(config-cgv6)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.

Step 3	**service-inline** *interface-name* **Example:** `RP/0/RSP0/CPU0:router(config-cgv6)#Service-inline interface TenGigE0/0/0/0`	Configures the service-inline interface.

Step 4	**service-type map-e** *instance-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-cgv6) #service-type map-e map1`
	`RP/0/RSP0/CPU0:router(config-cgV6-map_e)#`
		Configures the service type keyword definition for CGv6 MAP-E application.

Step 5	**cpe-domain ipv4 prefix** *ipv4 adress/prefix* **cpe-domain ipv6 prefix** *ipv6 adress/prefix*
	Example:
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)#cpe-domain ipv4 prefix 120.2.1.0/24`
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)#cpe-domain ipv6 prefix 9020:da8:2::/48`
		Configures the IPv4 or IPv6 prefixes of the CPE domain parameter without service modules.

Step 6	**sharing-ration 256** **Example:** `RP/0/RSP0/CPU0:router(config-cgv6-map_e)#sharing-ratio 256`	Configures the port sharing ratio. The value for the port sharing ratio is 256.
Step 7	**contiguous-port 16** **Example:** `RP/0/RSP0/CPU0:router(config-cgv6-map_e)#contiguous-ports 16`	Configures the contiguous port. The value for the contiguous port is 16.
Step 8	**br-endpoint-adress** **Example:** `RP/0/RSP0/CPU0:router(config-cgv6-map_e)#br-endpoint-address 9020:da8:2:ffff::1`	Configures the br-endpoint-address.

Step 9	**end**or **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)# end`
	`or`
	`RP/0/RSP0/CPU0:router(config-cgv6-map_e)# Commit`
		Saves configuration changes.
		• When you issue the end command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)?`
		`[cancel]:`

	Command or Action	Purpose
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

MAP-T without Service Cards

The objective of this feature is to ensure that the Mapping of Address and Port-Translation Mode (MAP-T) CGN solution can be implemented without service cards (VSM/ISM). With this feature, the CGN application directly interacts with the line cards to configure the MAP-T parameters and eliminates the dependency on the service cards.

Note The MAP-T CGN solution without service cards (VSM/ISM) is supported on Cisco IOS XR and Cisco IOS XR 64 bit operating system.

Restrictions for Configuring MAP-T without Service Card on Cisco IOS XR 32-bit Operating System

- MAP-T is supported on Cisco ASR 9000 Series and Cisco ASR 9900 Series 3rd, 4th, and 5th generation Ethernet line cards.
- If this feature is enabled on an interface, other PBR (policy based routing) features such as CLI PBR, BGP Flow Spec, One Platform Kit (onePK) or OpenFlow may not be functional; this is because only one PBR policy will be allowed on the interface.
- In a router, only one mode of either inline-service with service card or inline-service without service card will be supported.
- At a router level, the max scale limit for CPE domain parameters is 25 and for external domain parameters is 8k. If a single MAP-T instance has utilized 25 of CPE domain and 8k of external domain parameters; it is not possible to configure additional CPE and external domain parameters in the same router.

Restrictions for Configuring MAP-T without Service Card on Cisco IOS XR 64-bit Operating System

- From Cisco IOS XR Release 7.0.1 onwards, Cisco ASR 9000 Series 4th Generation Ethernet line cards support MAP-T.

- From Cisco IOS XR Release 7.1.2 onwards, Cisco ASR 9000 Series 5th Generation Ethernet line cards support MAP-T.
- MAP-T can be enabled in normal unicast routing scenario with default VRF. With non-default VRF, MAP-T works only in L3VPN and 6VPE cloud.
- For the IPv6 prefix length greater than 48 and less than 64 the sharing-ratio and contiguous-ports configurations are not considered during the translation.
- The Exception and Fragmented packets are not supported with inline MAP-T.
- In a single MAP-T instance only 255 CPE-domains are supported.



Note

- When you configure the MAP-T on Cisco ASR 9000 fourth generation line cards, verify the value in the num free field of the TCAM table.

  To verify the num free field values, use the **show controller controllers rm tcam summary 640-ING all np all location <node-id>** command.

  The number of MAP-T external domains that you can configure is based on the num free field values. When you configure a number of MAP-T external domains that exceed the num free field value, it results in a complete loss of traffic.

- Adding duplicate prefixes that already exist in other domains is not supported. If attempted, the RIB updates to map these prefixes to the new domain. If the duplicate domain is then deleted, the RIB removes the prefix instead of reverting it to the original domain, which interrupts forwarding.

This table shows the supported values for configuring MAP-T feature:

Table 4: Maximum Supported Values for Configuring MAP-T without Service Card on Cisco IOS XR 64-bit

Parameters	Maximum Supported Values
CGv6 Services	6
MAP-T Instances	255
CPE-domain	1023
External-domains	8191

Configuring MAP-T without Service Cards

To configure a MAP-T without service cards, perform the steps below.

SUMMARY STEPS

1. **configure**
2. **service cgv6** *instance-name*
3. **service-inline interface** *type interface-path-id*
4. **service-type map-t-cisco***instance-name*

5. **cpe-domain ipv4 prefix length** *value*
6. **cpe-domain ipv6 vrf** *vrf-name*
7. **cpe-domain ipv6 prefix length** *value*
8. **sharing ratio***number*
9. **contiguous-ports***number*
10. **cpe-domain-name** *cpe-domain-name* **ipv4 prefix** *address/prefix* **ipv6 prefix** *address/prefix*
11. **ext-domain-name** *ext-domain-name* **ipv6 prefix** *address/prefix* **ipv4-vrf** *vrf-name*
12. Use the **commit** or **end** command.
13. **show policy-map transient type pbr**
14. **show pbr service-node table summary**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**service cgv6** *instance-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# service cgv6 cgv6-1`
	`RP/0/RP0/CPU0:router(config-cgv6)#`
		Configures the instance for the CGv6 application and enters CGv6 configuration mode.
		Note
		The maximum number of CGv6 applications allowed for a MAP-T instance is 6.

Step 3	**service-inline interface** *type interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router(config-cgv6)# service-inline interface TenGigE0/0/0/0`
	`RP/0/RP0/CPU0:router(config-cgv6)#`
		Specifies an Ethernet interface on which the CGv6 service must be enabled.

Step 4	**service-type map-t-cisco***instance-name* **Example:** `RP/0/RP0/CPU0:router(config-cgv6)# service-type map-t-cisco map1`	Configures the service type keyword definition for CGv6 MAP-T application. application.
Step 5	**cpe-domain ipv4 prefix length** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-map-t-cisco)#` **`cpe-domain ipv4 prefix length 24`**	Configures the IPv4 prefix of the CPE domain
Step 6	**cpe-domain ipv6 vrf** *vrf-name* **Example:** `RP/0/RP0/CPU0:router(config-cgn-map-t-cisco)#` **`cpe-domain ipv6 vrf mapt-v6`**	Enables Virtual Routing and Forwarding (VRF) for the MAP-T configuration.

	Command or Action	Purpose
Step 7	**cpe-domain ipv6 prefix length** *value* **Example:** `RP/0/RP0/CPU0:router(config-cgn-map-t-cisco)#cpe-domain ipv6 prefix length 48`	Assigns a value for the ipv6-prefix length to be used as part of the MAP-T instance.
Step 8	**sharing ratio***number* **Example:** `RP/0/RP0/CPU0:router(config-cgn-map-t-cisco)#sharing-ratio 256`	Enters global configuration mode.
Step 9	**contiguous-ports***number* **Example:** `RP/0/RP0/CPU0:router(config-cgn-map-t-cisco)#contiguous-ports 8`	Specifies the Port Set ID (PSID) configuration.
Step 10	**cpe-domain-name** *cpe-domain-name* **ipv4 prefix** *address/prefix* **ipv6 prefix** *address/prefix* **Example:** `RP/0/RP0/CPU0:router(config-cgn-map-t-cisco)#cpe-domain-name cpe1 ipv4-prefix 10.0.0.1 ipv6-prefix 1000:1000::1`	Configures IPv4 and IPv6 prefix for a specific CPE domain.
Step 11	**ext-domain-name** *ext-domain-name* **ipv6 prefix** *address/prefix* **ipv4-vrf** *vrf-name* **Example:** `RP/0/RP0/CPU0:router(config-cgn-map-t-cisco)#ext-domain-name ext1 ipv6-prefix 2000:2000::1/48 ipv4-vrf mapt`	Configures IPv6 prefix and IPv4 VRF for the external domain.
Step 12	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 13	**show policy-map transient type pbr** **Example:** `RP/0/RP0/CPU0:router#show policy-map transient type pbr`	Displays the transient list type pbr of the policy-map.
Step 14	**show pbr service-node table summary** **Example:**	Displays the output for the class-maps.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router#show pbr service-node table summary`	

Configuration Example

Running Configuration

Verification

The following example shows the configuration of MAP-T without service cards:

```
RP/0/RP0/CPU0:router#show running-config service cgv6 cgv6-1
service cgv6 cgv6-1
service-inline interface Bundle-Ether2
service-type map-t-cisco mapt1
  cpe-domain ipv4 prefix length 24
  cpe-domain ipv6 vrf SVRF-003
  sharing-ratio 256
  contiguous-ports 8
  cpe-domain-name cpe1 ipv4-prefix 192.1.1.0 ipv6-prefix 2301:d01:1122::
  ext-domain-name ext1 ipv6-prefix 3301:d01:1122::/48 ipv4-vrf VRF-1
!
!
```

The following example shows the running configuration of MAP-T without service cards:

The following example shows the verification output:

```
RP/0/RP0/CPU0:router#show policy-map transient type pbr
policy-map type pbr CGN_0
handle:0x38000002
table description: L3 IPv4 and IPv6
class handle:0x78000003  sequence 1
   match destination-address ipv4 192.1.1.0 255.255.255.0--->should match the cpe domain
IPV4 address and mask
  punt service-node index 1001 app-id 0 local-id 0xfa1
!
 class handle:0x78000004  sequence 1
   match destination-address ipv6 3301:d01:1122::/48--->should match the ext domain IPV6
address and mask
  punt service-node index 2001 app-id 0 local-id 0x1771
!
 class handle:0xf8000002  sequence 4294967295 (class-default)
!
 end-policy-map
```

The following example shows the output for the class-maps:

```
RP/0/RP0/CPU0:router#show pbr service-node table summary

Service node count: 4
nodeid node0_RSP1_CPU0
_____________________________________
 Name          VIdx          Enc
_____________________________________
 CGN_1001           1001   cgn
 CGN_3001           3001   cgn
 CGN_5001           5001   cgn
 CGN_7001           7001   cgn
```

MAP-T Enhancements

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
MAP-T Enhancements	Release 7.11.1	Using Mapping of Address and Port Translation (MAP-T), you can now configure the IPv6 CPE domain prefix length as a non-multiple of eight, which enhances the prefix pool to accommodate a flexible size of the network portion of the IPv6 address. MAP-T also supports non-TCP/UDP/ICMP packets, ICMP error messages, and fragmented packets, which ensure reliable IPv6 connectivity over an IPv4 infrastructure and the other way around. These enhancements are available only on Cisco ASR 9000 Series Fifth Generation high-density Ethernet line cards. The feature introduces these changes: **CLI**: New • service-ipv4-mtu • service-ipv6-mtu • service-ipv4-nexthop addr • service-ipv6-nexthop addr commands. **YANG Data Model**: Cisco-IOS-XR-se-cgn-cfg (see GitHub, YANG Data Models Navigator)

From Release 7.11.1, the Mapping of Address and Port Translation (MAP-T) on Cisco ASR 9000 Series Fifth Generation high-density ethernet line cards supports:

- Configuring the IPv6 CPE domain prefix length as a nonmultiple of eight
- MAP-T of non-TCP/UDP/ICMP packets
- MAP-T of ICMP error messages

L4 Packets (Non-TCP/UDP/ICMP) and ICMP Error Messages Handling

Figure 3: MAP-T L4 packets (Non-TCP/UDP/ICMP) and ICMP Error Messages Handling Topology



1. On a dedicated interface HundredGigE0/0/0/1, when a Cisco ASR 9000 Series Router which acts as Border-Relay (BR) receives the IPv6 packets (Listed in the Table: *MAP-T Supported Packets Types*) from

R1 router over an IPv6 network, BR performs MAP-T of the packet from IPv6 to IPv4 and forwards to R2 router on HundredGigE0/3/0/1 interface over an IPv4 network.

2. When BR receives the IPv4 packets (Listed in the Table: *MAP-T Supported Packets Types*) from R2 router on a dedicated interface HundredGigE0/3/0/1 over an IPv4 network, it performs MAP-T of the packet from IPv4 to IPv6 and forwards to R1 router on HundredGigE0/0/0/1 over an IPv6 network.

Table 6: MAP-T Supported Packets Types

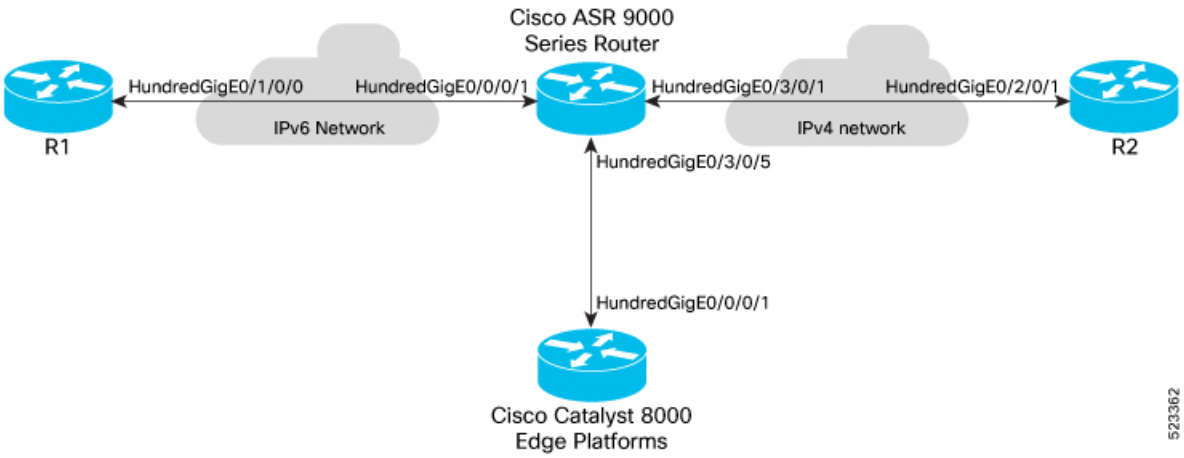
Packet Type	Description
Layer-4 packets	Layer-4 packets such as GRE, ESP, AH, IP-in-IP, L2TP, SCTP
ICMP Type 11	Time exceeded Message
ICMP Type 3 Code 0—4 messages	ICMP Type 3 (destination Unreachable) message supports the following codes: • Code—0: Net unreachable • Code—1: Net unreachable • Code—2: Protocol Unreachable • Code—3: Port Unreachable • Code—4: Fragmentation needed and DF set **Note** An ICMPv4 type 3 code 4 message is generated by Cisco ASR 9000 Series Router acting as BR, when: • The incoming packet size is larger than the configured IPv6 service MTU value. • DF bit is set to 1.

Fragmentation Handling

Prerequisite:

Ensure that you have any router from the Cisco Catalyst 8000 edge platforms family to handle the translation of fragmented packets whose length is greater than the configured service MTU.

Figure 4: MAP-T Fragmented Traffic Handling Topology



On a dedicated ingress inline interface, when a Cisco ASR 9000 Series Router receives a packet from R2 router on the HundredGigE0/3/0/1 interface, whose length is larger than the configured service IPv6 MTU size, it requires fragmentation. The received packet is sent to the Cisco Catalyst edge router to handle the translation of the fragmented packet on the HundredGigE0/3/0/5 interface. If the incoming packet from router R2 itself is fragmented packet, the fragmented packet is sent to the Cisco Catalyst 8000 edge router to handle its MAP-T. After the fragmentation is handled at the Cisco Catalyst edge router, the fragmented packet is sent back from the Cisco Catalyst edge router to the Cisco ASR 9000 Series Router on the HundredGigE0/3/0/5 interface and the checksum is calculated for the fragmented packet. The Cisco ASR 9000 Series Router forwards these packets to R1 router on the HundredGigE0/1/0/0 interface over an IPv6 network.

Note The Cisco Catalyst edge router can be replaced with any other router which can handle MAP-T of fragmented packets and is in compliant with MAP-T standards.

For details on Cisco Catalyst 8000 edge router configuration for fragmented packets, see Cisco Catalyst 8000 Edge Platform Router Configuration, on page 153

From Release 7.11.1, the following ICMP error messages types are translated.

Table 7: Translating ICMP IPv4 Headers to ICMP IPv6 Headers

Error Type	ICMP IPv4 Header	Translated ICMP IPv6 Header	Description
Time exceeded (Type 11)	Time exceeded (Type 11)	Type 3	Time Exceeded

Error Type	ICMP IPv4 Header	Translated ICMP IPv6 Header	Description
Destination Unreachable (Type 3)	Network Unreachable (code 0)	Type 1 code 0	Communication with destination administratively prohibited
	Host Unreachable (code 1)	Type 1 code 0	No route to destination
	Protocol Unreachable (code 2)	Type 4 code 1	Unrecognized Next Header type encountered
	Port Unreachable (code 3)	Type 1 code 4	Port unreachable
	Fragmentation Needed and DF was set (code 4)	Type 2 code 0	This message is translated to ICMPv6 **Packet Too Big** message (Type 2) with code set to zero. When the configured CGN service MTU values are: • Zero—MTU value is set to the default value which is 1280 bytes. • Greater than zero—MTU value is set to 1280+20 bytes. • Not configured for path MTU discovery (PMTUD) noncompliance system—Packet is dropped.

Table 8: Translating ICMP IPv6 Headers to ICMP IPv4 Headers

Error Type	ICMP IPv6 Header	Translated ICMP IPv4 Header	Description
Time Exceeded (type 3)	Hop limit exceeded in transmit (code 0)	Type 11 code 0	Time to Live exceeded in Transit

Error Type	ICMP IPv6 Header	Translated ICMP IPv4 Header	Description
Destination Unreachable (Type 1)	No route to destination (code 0)	Type 3 Code 1	No route to destination
	Communication with destination prohibited (code 1)	Type 1 code 10	Communication with destination prohibited
	Beyond scope of source address (code 2)	Type 1 Code 1	Beyond scope of source address
	Address unreachable (code 3)	Type 1 Code 1	Address unreachable
	Port unreachable (code 4)	Type 1 code 3	Port unreachable
	Packet Too Big (Type 2)	Type 3 code 4	**Fragmentation Needed** and **Don't Fragment** bit is set. When the configured CGN service MTU values are: • Zero—MTU value defined in the Packet Too Big Message is considered. • Other than zero—MTU value is set as the MTU value in the Packet Too Big Message-20 bytes. • Not configured for path MTU discovery (PMTUD) noncompliance system—Packet is dropped.

Limitations and Scale Statistics

Limitations

- MAP-T logging isn't supported for fragmented packets. You can view the fragmented packet count using the **show cgv6 map-t-cisco** *map-t-name***statistics** command.
- Ensure that the nexthop is reachable from the Cisco ASR 9000 Series Router through the LSP where MAP-T is configured.

  When there's no route or adjacency to redirect nexthop or the link to the second device (Catalyst 8000 Edge Platforms) is down, all packets are forwarded to the existing IPv4 or IPv6 exception path and dropped in LSP NPU. The dropped packets aren't visible in the drop counters of the **show cgv6 map-t-cisco** *map-t-name* **statistics** command, but they appear under a **set to redirect** counters of the same command.

- Any modification to the existing MAP-T instance configuration clears all the statistics of that instance.
- MAP-T supports configuring the IPv6 CPE domain prefix length as a nonmultiple of eight on Cisco ASR 9000 Series Fifth Generation High-Density Ethernet line cards. For other generation line cards, the router accepts the configuration but logs an error message in syslog as

```
Router# non-byte order CPE Domain Prefix len:41 is not supported on this Linecard.
```

Scale Statistics

ASR 9000 Series Router support:

- 255 MAP-T instances per system
- 255 CPE domains per MAP-T instance
- 1023 CPE domains per system
- 8191 external-domains per system
- 6 CGv6 services per system

Configuring MAP-T Enhancements

Configuring MAP-T enhancement on Cisco ASR 9000 Series Fifth Generation High-Density Ethernet Line Cards includes:

- MAP-T Configuration on Cisco ASR 9000 Series Router, on page 147
- Cisco Catalyst 8000 Edge Platform Router Configuration, on page 153

MAP-T Configuration on Cisco ASR 9000 Series Router

Configuration Example

```
Router# configure
Router(config)# service cgv6 cgn6
Router(confi-cgv6)# service-inline interface Bundle-Ether1
Router(confi-cgv6)# service-type map-t-cisco maptff
/*maptff — MAP-T instance name*/
Route(confi-cgv6-mapt-cisco)# cpe-domain ipv4 prefix length 30
Router(confi-cgv6-mapt-cisco)# cpe-domain ipv6 vrf default
Router(confi-cgv6-mapt-cisco)# cpe-domain ipv6 prefix length 56
Router(confi-cgv6-mapt-cisco)# sharing-ratio 64
Router(confi-cgv6-mapt-cisco)# contiguous-ports 16
Router(confi-cgv6-mapt-cisco)# cpe-domain-name cpe5 ipv4-prefix 192.0.2.1 ipv6-prefix
2001:db8:0002:100::/48
Router(confi-cgv6-mapt-cisco)# ext-domain-name ext5 ipv6-prefix 2001:DB8::/48 ipv4-vrf
default

/* Use the following configuration to configure MTU and for traffic redirection/*
Router(confi-cgv6-mapt-cisco)# service-ipv4-mtu 1300
Router(confi-cgv6-mapt-cisco)# service-ipv6-mtu 1294
Router(confi-cgv6-mapt-cisco)# service-ipv4-nexthop addr 8.8.8.2 vrf default
Router(confi-cgv6-mapt-cisco)# service-ipv6-nexthop addr 8:8:8::2 vrf default
Router(confi-cgv6-mapt-cisco)# commit
Router(confi-cgv6-mapt-cisco)# end
```

Running Configuration

```
service cgv6 cgn6
 service-inline interface Bundle-Ether1
 service-type map-t-cisco maptff
  cpe-domain ipv4 prefix length 30
  cpe-domain ipv6 vrf default
  cpe-domain ipv6 prefix length 56
  sharing-ratio 64
  contiguous-ports 16
  service-ipv4-mtu 1300
  service-ipv6-mtu 1294
  service-ipv4-nexthop addr 8.8.8.2 vrf default
  service-ipv6-nexthop addr 8:8:8::2 vrf default
  cpe-domain-name cpe5 ipv4-prefix 192.0.2.1 ipv6-prefix 2001:db8:0002:100::/48
  ext-domain-name ext5 ipv6-prefix 2001:DB8::/48 ipv4-vrf default
 !
!
```

Note **VRF** configuration is optional. When no VRF is configured, the default VRF is considered for redirecting the packets using service IPv4 or IPv6 nexthop address.

Statistics of MAP-T Functionality

Use the **sh int bundle-ether 1** command to check the bundle ether interface configuration.

```
Router# sh int bundle-ether 1
Bundle-Ether1 is down, line protocol is down
  Interface state transitions: 0
  Hardware is Aggregated Ethernet interface(s), address is 0024.f71f.790d
  Internet address is Unknown
  MTU 1514 bytes, BW 0 Kbit
     reliability 255/255, txload Unknown, rxload Unknown
  Encapsulation ARPA,
  Full-duplex, 0Kb/s
  loopback not set,
    No. of members in this bundle: 2
      HundredGigE0/0/0/12          Full-duplex  100000Mb/s   Configured
      HundredGigE0/0/0/13          Full-duplex  100000Mb/s   Configured
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets

Router# sh contr np ports  all location 0/0/CPU0

                Node: 0/0/CPU0:
----------------------------------------------------------------

NP Bridge Fia                       Ports
-- ------ --- -----------------------------------------------
0  --     0   TenGigE0/0/0/0/0 - TenGigE0/0/0/0/3, HundredGigE0/0/0/1 - HundredGigE0/0/0/3
```

```
1  --     1   HundredGigE0/0/0/4 - HundredGigE0/0/0/7
2  --     2   HundredGigE0/0/0/8 - HundredGigE0/0/0/11
3   --     3   HundredGigE0/0/0/12 - HundredGigE0/0/0/14, TenGigE0/0/0/15/0 -
TenGigE0/0/0/15/3
```

NP number **3** is for both the bundle members. Use the **sh contr np counters np3 location 0/0/CPU0** command to see NP3 MAP-T counters.

```
Router# sh contr np counters np3 location 0/0/CPU0

               Node: 0/0/CPU0:
----------------------------------------------------------------

Show global stats counters for NP3, revision v0

Last clearing of counters for this NP: NEVER

Read 0 non-zero NP counters:
Offset  Counter                                                FrameValue   Rate (pps)
--------------------------------------------------------------------------------------
 117  Drop packets captured                                        7255           0
 227  L2 aging scan not enabled                                    3556           0
 289  MAP-T Service Cfg Common Lkp No Match                          19           0
 290  MAP-T Service Cfg Meta Lkp No Match                            41           0
 291  MAP-T Service Ext Common Lkp No Match                           1           0
 292  MAP-T Service Ext Meta Lkp No Match                            26           0
 293  MAP-T Service Processing Entry                         3578614234        1011
 294  MAP-T frag pkt set to redirect for transl                16163049        1011
 295  MAP-T v4 pkt set to redirect for transl                    118331        1011
 296  MAP-T v4 pkt set to redirect for v6 frag needed            101174           0
 298  MAP-T v4 to v6 translated                                   65233           0
 299  MAP-T v6 pkt set to redirect for transl                  25223172           0
 300  MAP-T v6 pkt set to redirect for v4 frag needed           9077280           0
 301  MAP-T v6 to v4 translated                              3549122635           0
 308  MAPT - TBPG Event                                        1060386         171
 367  TBPG L2 mailbox events                                    933891         150
 368  TBPG MAC scan events                                       15568           3
 369  TBPG stat events                                         59769049        9626
 433  Egress UIDB in down state                                    3           0
 650  MAP-T v4 to v6 drops                                        87           0
 651  MAP-T v6 to v4 drops                                   4084776           0
 784  Inject to port                                             282           0
1430  IPv6 disabled in UIDB                                   353218           0
1503  ARP                                                          1           0
1524  Diags                                                       96           0
1611  IFIB                                                         5           0
1695  Diags RSP active                                            99           0
1755  IPv4 adjacency null route                                   36           0
2001  IPv4 incomplete Tx adjacency                               220           0
2002  Punt policer: IPv4 incomplete Tx adjacency               15549           0
2016  IPv6 incomplete Tx adjacency                               216           0
2017  Punt policer: IPv6 incomplete Tx adjacency                2161           0
  HW  Received from Line                                  3578967506        1011
  HW  Transmit to Fabric                                  3574529482        1011
  HW  Received from Fabric                                3574294217           0
  HW  Transmit to Line                                    3574276238           0
  HW  Host Inject Received                                      365           0
  HW  Host Punt Transmit                                        574           0
  HW  Local Loopback Received at iGTR                           407           0
  HW  Local Loopback Transmit by iGTR                           407           0
  HW  Local Loopback Received at Egress                         407           0
  HW  Transmit to TM from eGTR                            3574276911           0
  HW  Transmit to L2                                      3574276812           0
  HW  Received from Service Loopback                            99           0
```

```
  HW  Transmit to Service Loopback                                  99           0
  HW  Internal generated by PDMA                                145853497       23477
```



Note

- Use the **sh contr np ports all location <loc>** command to view the network processor's global MAP-T packet count.
- Use the **clear cgv6 map-t-cisco mapt statistics** command to clear the MAP-T statistics.

MAP-T Instance Counters

Use the **sh cgv6 map-t-cisco** *map-t instance name* **mapt statistics** command to view the counters specific to a MAP-T instance. MAP-T instance is configured using a **service-type map-t-cisco** command. MAP-T instance-specific counters along with their description can be found in the MAP-T Instance Counters and Description table.

```
Router# sh cgv6 map-t-cisco mapt statistics
Map-t-cisco IPv6 to IPv4 counters:
...
...
Translated other L4 protocol Count: 0

Udp Fragmentation Needed Packet set to Redirect Count: 0

Udp Fragmentation Needed Packet Drop before Redirection Count : 0

Udp Fragmented Packet set to Redirect Count: 0

Udp Fragmented Packet Drop before Redirection Count: 0

Tcp Fragmentation Needed Packet set to Redirect Count: 0

Tcp Fragmentation Needed Packet Drop before Redirection Count : 0

Tcp Fragmented Packet set to Redirect Count: 0

Tcp Fragmented Packet Drop before Redirection Count: 0

Icmp Fragmentation Needed Packet set to Redirect Count: 0
Icmp Fragmentation Needed Packet set to Redirect Count: 0

Icmp Fragmented Packet set to Redirect Count: 0

Icmp Fragmented Packet Drop before Redirection Count : 0

Other Fragmentation Needed Packet set to Redirect Count: 0

Other Fragmentation Needed Packet Drop before Redirection Count : 0

Other Fragmented Packet set to Redirect Count: 0

Other Fragmented Packet Drop before Redirection Count : 0
...
...
Map-t-cisco IPv4 to IPv6 counters:
...
...
Translated other L4 protocol Count: 0
```

```
Udp Fragmentation Needed Packet set to Redirect Count: 0

Udp Fragmentation Needed Packet Drop before Redirection Count : 0

Udp Fragmented Packet set to Redirect Count: 0

Udp Fragmented Packet Drop before Redirection Count: 0
Tcp Fragmentation Needed Packet set to Redirect Count: 0

Tcp Fragmentation Needed Packet Drop before Redirection Count : 0

Tcp Fragmented Packet set to Redirect Count: 0

Tcp Fragmented Packet Drop before Redirection Count: 0

Icmp Fragmentation Needed Packet set to Redirect Count: 0

Icmp Fragmentation Needed Packet set to Redirect Count: 0

Icmp Fragmented Packet set to Redirect Count: 0

Icmp Fragmented Packet Drop before Redirection Count : 0

Other Fragmentation Needed Packet set to Redirect Count: 0

Other Fragmentation Needed Packet Drop before Redirection Count : 0

Other Fragmented Packet set to Redirect Count: 0

Other Fragmented Packet Drop before Redirection Count : 0
```

The following table lists the MAP-T instance counters on Cisco ASR 9000 Series Fifth Generation High-Density Ethernet line cards and their description.

Table 9: MAP-T Instance Counters and Description

Counter	Description
`Translated other L4 protocol Count`	Nonstandard L4 protocol packet (Non-TCP/UDP/ICMP packets) that are translated by NPU.
`Udp Fragmentation Needed Packet set to Redirect Count`	UDP packets having packet size larger than the configured service MTU value, which is set to redirect to the second device.
`Udp Fragmentation Needed Packet Drop before Redirection Count`	UDP packets having packet size larger than the configured service MTU value, which need to be redirected to the second device, but are dropped due to Port Set Identifier (PSID) check.
`Udp Fragmented Packet set to Redirect Count`	UDP fragmented packets, which are set to redirect to the second device.
`Udp Fragmented Packet Drop before Redirection Count`	UDP fragmented packets, which need to be redirected to the second device, but are dropped due to PSID check.

Counter	Description
`Tcp Fragmentation Needed Packet set to Redirect Count`	TCP packets having packet size larger than the configured service MTU value, which is set to redirect to the second device.
`Tcp Fragmentation Needed Packet Drop before Redirection Count`	TCP packets having packet size larger than the configured service MTU value, which needs to be redirected to the second device, but are dropped due to PSID check.
`Tcp Fragmented Packet set to Redirect Count`	TCP fragmented packets, which are set to redirect to the second device.
`Tcp Fragmented Packet Drop before Redirection Count`	TCP fragmented packets, which need to be redirected to the second device, but are dropped due to PSID check.
`Icmp Fragmentation Needed Packet set to Redirect Count`	ICMP packets having packet size larger than the configured service MTU value, which is set to redirect to the second device.
`Icmp Fragmentation Needed Packet Drop before Redirection Count`	ICMP packets having packet size larger than the configured service MTU value, which needs to be redirected to the second device, but are dropped due to PSID check.
`Icmp Fragmented Packet set to Redirect Count`	ICMP fragmented packets, which are set to redirect to the second device.
`Icmp Fragmented Packet Drop before Redirection Count`	ICMP fragmented packets, which need to be redirected to the second device, but are dropped due to PSID check.
`Other Fragmentation Needed Packet set to Redirect Count`	Nonstandard L4 protocol packet (Non-TCP/UDP/ICMP packets) having packet size larger than the configured service mtu value, which is set to redirect to the second device.
`Other Fragmentation Needed Packet Drop before Redirection Count`	Nonstandard L4 protocol packet (Non-TCP/UDP/ICMP packets) having packet size larger than the configured service mtu value, which needs to be redirected to the second device, but are dropped due to PSID check.
`Other Fragmented Packet set to Redirect Count`	Nonstandard L4 protocol packet (Non-TCP/UDP/ICMP packets) fragmented packets, which are set to redirect to the second device.
`Other Fragmented Packet Drop before Redirection Count`	Nonstandard L4 protocol packet (Non-TCP/UDP/ICMP packets) fragmented packets, which need to be redirected to the second device, but are dropped due to PSID check.

Cisco Catalyst 8000 Edge Platform Router Configuration

Use the following configuration in a redirection router (Cisco Catalyst 8000 Edge Platforms) to handle fragment traffic.

Configuration Example

```
Router# nat64 map-t domain 3
/*Domain range is from 1-10000*/
Router(config-nat64-mapt)# default-mapping-rule Ip-v6-prefix 2001:DB8::/48
/*Default Mapping Rule maps to IPv6 prefix in ext-domain configuration*/
Router(config-nat64-mapt)# basic-mapping-rule
/*Basic Mapping Rule maps to IPv6 prefix in cpe-domain configuration.*/
Router(config-nat64-mapt-bmr)# ip-v6-prefix 2001:db8:0002:100::/48
Router(config-nat64-mapt-bmr)# ip-v4-prefix 192.0.2.1/24
Router(config-nat64-mapt-bmr)# port-parameters share-ratio 256 start-port 1024
```



Note **contiguous-ports** isn't a configurable parameter. It's derived as follows:

```
contiguous-ports = start-port / share-ratio.
```

When **start-port** is 1024 and **share-ratio** is 256, the **contiguous-ports** value is 4.

Running Configuration

```
MAP-T Domain 3
   Mode MAP
   Default-mapping-rule
      Ip-v6-prefix 2001:DB8::/48
   Basic-mapping-rule
      Ip-v6-prefix 2001:db8:0002:100::/48
      Ip-v4-prefix 192.0.2.1/24
      Port-parameters
         Share-ratio 256   Contiguous-ports 4   Start-port 1024
         Share-ratio-bits 8   Contiguous-ports-bits 2   Port-offset-bits 7
```

Verification

```
Router# sh nat64 statistics
NAT64 Statistics

Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Sessions found: 1
Sessions created: 1
Expired translations: 0
Global Stats:
          ....
          ....
          ....
         MAP-T: 612970
         MAP-E: 0
```

TCP MSS for MAP-T

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
Configurable TCP Maximum Segment Size for MAP-T	Release 24.2.1	You can now configure a TCP Maximum Segment Size (MSS) for each MAP-T instance and adjust the MSS value of the forwarded Synchronize (SYN) packets to the configured value. This ensures that TCP segments are within the network capacity and prevents fragmentation. The feature introduces these changes: **CLI:** • **tcp mss (map-t)** **YANG Data Model**: Cisco-IOS-XR-um-service-cfg (see GitHub, YANG Data Models Navigator)

The TCP Maximum Segment Size (MSS) defines the largest segment of data that a device can handle in a single TCP packet, excluding the TCP and IP headers. You can now configure an MSS for each MAP-T instance and adjust the MSS value of the forwarded Synchronize (SYN) packets to the configured value.

Configuring the MSS to match or be smaller than the Maximum Transmission Unit (MTU) of the path helps prevent fragmentation and ensures efficient network performance by aligning the segment size with the path MTU.

This feature is supported on Cisco ASR 9000 Series 5th Generation Ethernet Line Cards:

Configure TCP MSS for MAP-T

Optimize TCP segment size to enhance IPv4 and IPv6 translation.

Procedure

Step 1 Configure the instance for the CGv6 application.

Example:

```
Router# configure
Router(config)# service cgv6 test-service
Router(confi-cgv6)#
```

Note
The maximum number of CGv6 applications allowed for a MAP-T instance is 6.

Step 2 Configure the service-inline interface and service type for the CGv6 MAP-T application.

Example:

```
Router(config-cgv6)# service-inline interface Bundle-Ether1
Router(config-cgv6)# service-type map-t-cisco mapt
Router(config-cgv6-mapt-cisco)#
```

Step 3 Configure TCP MSS for IPv4 and IPv6 address-family.

Example:

```
Router(config-cgv6-mapt-cisco)# address-family ipv6
Router(config-cgn-mapt-cisco-afi)# tcp mss 1400
Router(config-cgv6-mapt-cisco)# address-family ipv4
Router(config-cgn-mapt-cisco-afi)# tcp mss 1400
Router(config-cgn-mapt-cisco-afi)# commit
```

Step 4 Verify your service CGv6 configuration by running the **show running-config service cgv6** command.

Example:

```
RP/0/RSP0/CPU0:ios# show running-config service cgv6 test-service
Tue Jun 11 08:50:05.174 UTC
service cgv6 test-service
 service-inline interface Bundle-Ether1
 service-type map-t-cisco mapt
  address-family ipv6
   tcp mss 1400
  !
  address-family ipv4
   tcp mss 1400
  !
 !
!
```

Scaling Up of MAP-T Instances

Table 11: Feature History Table

Feature Name	Release Feature Information	Description
Scaling Up of MAP-T Instances	Release 24.2.1	You can now service an increased number of address-mapping requirements for the network traffic requiring a transition from IPv4 to IPv6 and IPv6 to IPv4 networks. Such scaling is possible because we've expanded our mapping of address and port translation (MAP-T) instance capacity increasing 255–2000, enabling more comprehensive address mapping for network traffic transitioning between IPv4 and IPv6. This increase improves scalability, load balancing, IPv4 address efficiency, and ensures seamless connectivity with enhanced performance and security across IPv4, IPv6 networks. These expansions are available only on Cisco ASR 9000 Series 5th Generation High-Density Multi-Rate Line Cards. The feature introduces these changes: **CLI:** • **hw-module profile itcam lightspeed**

MAP-T (Mapping of Address and Port with Translation) supports communication between both IPv4 and IPv6 networks. This technology facilitates the data transmission from IPv4 to IPv6 and from IPv6 to IPv4, ensuring compatibility and connectivity between the devices and the networks running on these two different versions of the Internet Protocol. MAP-T allows IPv6-only networks to communicate with IPv4 networks. MAP-T combines address mapping and port translation, enabling multiple IPv6 hosts to share a single IPv4 address while distinguishing their traffic through unique port numbers. For more information on MAP-T, see MAP-T Enhancements, on page 142, and for the MAP-T specification, see RFC75990.

Configure TCAM for MAP-T Traffic

You can carve Ternary Content-Addressable Memory (TCAM) space as per your incoming and outgoing IPv4, IPv6 entries to support MAP-T.

Procedure

Step 1 To configure TCAM blocks for the network's MAP-T traffic, use the **hw-module profile itcam lightspeed v4-ing 1 v4-eng 1 v6-ing 13 v6-egr 1 location** command.

Example:

```
Router# configure
Router(config)# hw-module profile itcam lightspeed v4-ing 1 v4-eng 1 v6-ing 13 v6-egr 1 location
```

```
0/6/CPU0
In order to activate this internal TCAM partition configuration, you must manually reload the line
card. This command must be used with caution and only when recommended by Cisco.
Router(config)# commit
```

Step 2 To verify the number of IPV4, IPv6 entries in TCAM, use the **show controller rm tcam sum all all np all loc** command.

Example:

```
Router# show controller rm tcam sum all all np all loc 0/6/cpu0 TCAM summary for NP0:

  TCAM Region: 160-ING (0)
  max entries: 16384, num free: 16384
    Feature ID: IPV4-ACL (0)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: L2-ACL (2)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: IPV4-LI (4)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: QOS-FMT-0 (8)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: QOS-FMT-1 (9)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: QOS-FMT-2 (10)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: QOS-FMT-3 (11)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: QOS-FMT-15 (13)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: PBR-IPV4 (14)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: PBR-MPLS (15)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: PBR-L2 (17)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: PBR-FMT-0 (18)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: IPV4-META-ACL (20)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
  TCAM Region: 160-EGR (1)
  max entries: 8192, num free: 8192
    Feature ID: IPV4-ACL (0)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: L2-ACL (2)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: QOS-FMT-0 (8)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
    Feature ID: QOS-FMT-1 (9)
      Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
```

```
0
   Feature ID: QOS-FMT-2 (10)
     Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
   Feature ID: QOS-FMT-3 (11)
     Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
   Feature ID: QOS-FMT-15 (13)
     Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
 TCAM Region: 640-ING (2)
 max entries: 1536, num free: 1536
   Feature ID: IPV6-ACL (1)
     Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
   Feature ID: IPV6-LI (5)
     Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
   Feature ID: EDPL (7)
     Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
   Feature ID: QOS-FMT-4 (12)
     Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
   Feature ID: PBR-IPV6 (16)
     Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
   Feature ID: PBR-FMT-4 (19)
     Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
   Feature ID: IPV6-META-ACL (21)
     Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
 TCAM Region: 640-EGR (3)
 max entries: 512, num free: 512
   Feature ID: IPV6-ACL (1)
     Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
   Feature ID: QOS-FMT-4 (12)
     Total vmr_ids per feature id: 0, Total used entries per feature id: 0 Total allocated entries:
0
```

Overview of MAP-T Logging

MAP-T Logging feature records and exports the IPv4 to IPv6 and IPv6 to IPv4 address translation information to the server. It captures all the following information and stores in the server as a template. It helps to map which IPv4 address translated to which corresponding IPv6 address, and vice versa.

A single translated flow captures the following details:

- IPv4 source address
- IPv4 destination address
- Source port
- Destination port
- VRF name configured in CPE-domain

- VRF name configured in EXT-domain
- Timestamp
- IPv6 source address
- IPv6 destination address

Following is the detail information:

- Ports are not being translated during MAP-T conversion so there is no pre-NAT or post-NAT display. Only one pair of IPv4 or IPv6 address or port is displayed in a MAP-T logging record.
- There is no indication that MAP-T converted flow is IPv4 to IPv6 or IPv6 to IPv4.
- The following information is displayed from the point of view IPv4 packet, which is IPv4 to IPv6:
  - The ingress VRF is of IPv4 address, and the egress VRF is of IPv6 address.
  - The Layer 4 destination port and source port are the same as seen in IPv4 packet header.
- Flows with same address pair in different VRFs are considered as separate flows.
- Any new flow that comes in, after 512k flows are learnt, is dropped. There is no MAP-T conversion for any new flow.
- Output interface is not displayed in the records.
- A 60 second timer runs by default for all flows. If any flow is active for 60 seconds, it is exported out. This is to ensure high availability. If a line card went down or there was a network process failure, all learned flow available in the system for at least 60 seconds will not be lost.

Restrictions

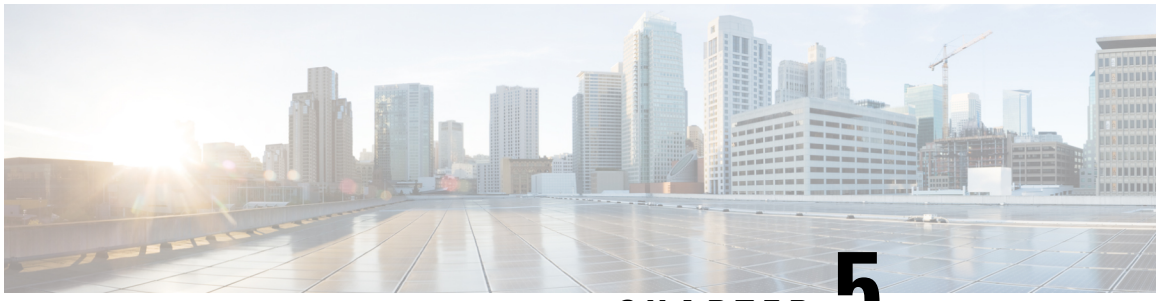
- Ipv4 to IPv6 and IPv6 to IPv4 address translation flow for same pair of address is considered as a single flow if IPv4 to IPv6 and IPvV6 to IPv4 traffic is on same network process. Flow learning across different network process cannot be considered as the same flow on that line card. If they are on a different network process, they are considered as two different flow. Flow collector application can identify unique flows.
- There is a 3-second deviation in flow expiration timer as it takes 3 seconds to scan all the 512k flows.

Configuration Example

```
/* Configure the MAP-T Monitor command */
RP/0/RSP0/CPU0:ios(config)# flow monitor-map map1
RP/0/RSP0/CPU0:ios(config-fmm)# record map-t
RP/0/RSP0/CPU0:ios(config-fmm)# exporter exp1

/* Apply the MAP-T monitor in the ingress interface where translation happens */
RP/0/RSP0/CPU0:ios(config)# interface HundredGigE 0/0/0/2
RP/0/RSP0/CPU0:ios(config-if)# flow map-t monitor map1 ingress

/* Configure the interface in Border Relay (BR) to export the flows to Logging server*/
RP/0/RSP0/CPU0:ios(config)# flow exporter-map exp1
RP/0/RSP0/CPU0:ios(config-fem)# version v9
RP/0/RSP0/CPU0:ios(config-fem)# source TenGigE0/4/0/2
/* Configure the IP address of the server interface */
RP/0/RSP0/CPU0:ios(config-fem)# destination 10.0.0.1
```



CHAPTER 5

Understanding MAP-T Exception Handling

The MAP-T Exception Handling with VSM feature handles fragmented packets, TCP MSS clamping, path MTU and ICMP packets. Depending on the header details of the fragmented packets, the CGN application directly interacts with the line cards to process or forward the packets to VSM.

Note When MAP-T is configured for exception handling with VSM, the static route details are automatically configured with Service App details for an IPV6 to IPv4 translated packet. No manual configuration is required to configure the static routes. But in the case of an IPv4 to IPv6 translated packet, static routes have to be manually configured.

- Configuration and Scalability Limits, on page 161
- Supported IPv6 Extension Headers, on page 162
- Processing of Fragmented Packets, on page 162
- Configuring MAP-T for Exception Handling with VSM, on page 163
- Running Configuration, on page 163
- Troubleshooting ICMP Errors, on page 165

Configuration and Scalability Limits

- Starting from Release 24.4.1, ICMP rate limiting is supported.
- Ensure that ServiceInfra is configured before other steps are executed to configure MAP-T for exception handling. The service infrastructure is used for the management plane.
- Configuring map-t-cisco instances :
  - Only a single external domain can be configured for each map-t-cisco instance.
  - The maximum number of CPE domains that can be configured for MAP-T exception handling is 1024.
  - The number of map-t-cisco instances supported for exception handling is 32 and the maximum number of CPE domain parameters that can be used in a single map-t-cisco instance is 256.
  - A map-t-cisco instance can be configured only with a default VRF.

- To ensure IPv4/IPv6 translation is successfully performed, set the path MTU of the tunnel for both IPv4 and IPv6 traffic to 1500 bytes.
- The IOS XR software does not support IPv6 fragmentation.
  - If an IPv4 packet that does not have a DF bit set is translated to an IPv6 packet that exceeds the configured MTU, the packet is translated by VSM and forwarded.

Note

- If the Path-MTU value is not configured, the default value is assumed to be 1280 bytes.
- Configure the MTU on physical VSM interfaces to 1518 bytes.

  - If an IPv6 packet that does not have a fragment header set is translated to an IPv4 packet that exceeds the minimum MTU, the packet is dropped and an ICMP error message is sent.

- Modification of path-mtu value is not recommended. If you modify the path-mtu setting, delete the service-inline interface and configure it again for the path-mtu changes to take effect.
- Deletion of ServiceApp configuration from a MAP-T domain is not permitted.

Supported IPv6 Extension Headers

This implementation of MAP-T exception handling supports the following IPV6 extension headers:

- Hop-by-Hop Options header (if this header exists, it must be the first one following the main/regular header)
- Destination Options header.
- Routing header.
- Fragment header.

Processing of Fragmented Packets

The following table shows how MAP-T handles a fragmented IPv6 packet that is translated to an IPv4 packet.

Fragment Header	More Fragment flag	Fragment off	Action taken
0	x	x	Lookup success. Processed by 8X100 GE line card
1	0	0	Lookup success based on protocol field. Forward packet to VSM

Fragment Header	More Fragment flag	Fragment off	Action taken
1	0	1	Last fragment. Forward packet to VSM.
1	1	0	First fragment. Forward packet to VSM
1	1	1	Middle fragment. Forward packet to VSM

The following table shows how MAP-T handles fragmentation for an IPv4 packet that is translated to an IPv6 packet.

Do Not Fragment bit	More Fragment bit	Fragment off	Action
0	0	0	Lookup success based on protocol value. The translated packet will contain a Fragment header
0	0	1	Lookup fail. Forward to VSM
0	1	0	Lookup fail. Forward to VSM
0	1	1	Lookup fail. Forward to VSM
1	0	0	Lookup success. Processed by 8X100 GE line card

Configuring MAP-T for Exception Handling with VSM

Perform these tasks to configure MAP-T for exception handling with VSM

Running Configuration

```
interface ServiceInfra1
ipv4 address 1.1.1.1 255.255.255.252
service-location 0/4/CPU0
!
interface ServiceApp1
ipv4 address 40.40.40.1 255.255.255.0
service cgv6 cgn123 service-type map-t-cisco
!
```

```
service cgv6 cgn123
service-location preferred-active 0/4/CPU0
service-inline interface tenGigE 0/0/0/0/2
service-type map-t-cisco map1
address-family ipv4
interface ServiceApp1
tcp mss 335
path-mtu 1200
!
address-family ipv6
interface ServiceApp2
tcp mss 1254
path-mtu 1500
!
cpe-domain ipv4 prefix length 24
cpe-domain ipv6 vrf default
cpe-domain ipv6 prefix length 48
sharing-ratio 256
contiguous-ports 16
cpe-domain-name cpe0 ipv4-prefix 192.1.1.0 ipv6-prefix 2301:0:1122::
cpe-domain-name cpe1 ipv4-prefix 192.1.2.0 ipv6-prefix 2301:1:1122::
cpe-domain-name cpe2 ipv4-prefix 192.1.3.0 ipv6-prefix 2301:2:1122::
cpe-domain-name cpe3 ipv4-prefix 192.1.4.0 ipv6-prefix 2301:3:1122::
cpe-domain-name cpe4 ipv4-prefix 192.1.5.0 ipv6-prefix 2301:4:1122::
cpe-domain-name cpe5 ipv4-prefix 192.1.6.0 ipv6-prefix 2301:5:1122::
cpe-domain-name cpe6 ipv4-prefix 192.1.7.0 ipv6-prefix 2301:6:1122::
cpe-domain-name cpe7 ipv4-prefix 192.1.8.0 ipv6-prefix 2301:7:1122::
cpe-domain-name cpe8 ipv4-prefix 192.1.9.0 ipv6-prefix 2301:8:1122::
cpe-domain-name cpe9 ipv4-prefix 192.1.10.0 ipv6-prefix 2301:9:1122::
cpe-domain-name cpe10 ipv4-prefix 192.1.11.0 ipv6-prefix 2301:a:1122::
cpe-domain-name cpe11 ipv4-prefix 192.1.12.0 ipv6-prefix 2301:b:1122::
cpe-domain-name cpe12 ipv4-prefix 192.1.13.0 ipv6-prefix 2301:c:1122::
cpe-domain-name cpe13 ipv4-prefix 192.1.14.0 ipv6-prefix 2301:d:1122::
cpe-domain-name cpe14 ipv4-prefix 192.1.15.0 ipv6-prefix 2301:e:1122::
```

```
cpe-domain-name cpe15 ipv4-prefix 192.1.16.0 ipv6-prefix 2301:f:1122::
cpe-domain-name cpe16 ipv4-prefix 192.1.17.0 ipv6-prefix 2301:10:1122::
cpe-domain-name cpe17 ipv4-prefix 192.1.18.0 ipv6-prefix 2301:11:1122::
cpe-domain-name cpe18 ipv4-prefix 192.1.19.0 ipv6-prefix 2301:12:1122::
cpe-domain-name cpe19 ipv4-prefix 192.1.20.0 ipv6-prefix 2301:13:1122::
cpe-domain-name cpe20 ipv4-prefix 192.1.21.0 ipv6-prefix 2301:14:1122::
cpe-domain-name cpe21 ipv4-prefix 192.1.22.0 ipv6-prefix 2301:15:1122::
cpe-domain-name cpe22 ipv4-prefix 192.1.23.0 ipv6-prefix 2301:16:1122::
cpe-domain-name cpe23 ipv4-prefix 192.1.24.0 ipv6-prefix 2301:17:1122::
cpe-domain-name cpe24 ipv4-prefix 192.1.25.0 ipv6-prefix 2301:18:1122::
cpe-domain-name cpe25 ipv4-prefix 192.1.26.0 ipv6-prefix 2301:19:1122::
cpe-domain-name cpe26 ipv4-prefix 192.1.27.0 ipv6-prefix 2301:1a:1122::
cpe-domain-name cpe27 ipv4-prefix 192.1.28.0 ipv6-prefix 2301:1b:1122::
cpe-domain-name cpe28 ipv4-prefix 192.1.29.0 ipv6-prefix 2301:1c:1122::
cpe-domain-name cpe29 ipv4-prefix 192.1.30.0 ipv6-prefix 2301:1d:1122::
cpe-domain-name cpe30 ipv4-prefix 192.1.31.0 ipv6-prefix 2301:1e:1122::
cpe-domain-name cpe31 ipv4-prefix 192.1.32.0 ipv6-prefix 2301:1f:1122::
ext-domain-name ext1 ipv6-prefix 6301:d01:1122::/48 ipv4-vrf default
!
```

Troubleshooting ICMP Errors

This section covers details of errors encountered during translation of packets from IPv4 to IPv6 and vice versa.

Translating ICMP IPv4 Headers to ICMP IPv6 Headers

Translating ICMP IPv6 Headers to ICMP IPv4 Headers

Translating ICMP IPv4 Headers to ICMP IPv6 Headers

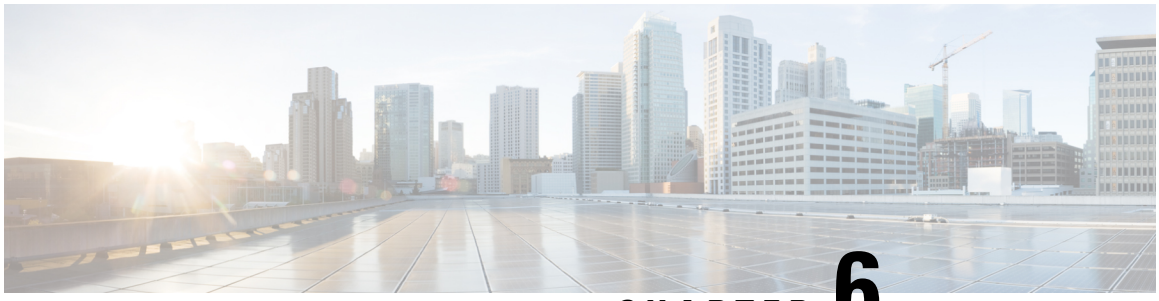
	IPv4 Header	IPv6 Header	Description of the error
Destination Unreachable (Type 3)	Network Unreachable (code 0)	Type 1 code 0	Communication with destination administratively prohibited
	Host Unreachable (code 1)	Type 1 code 0	No route to destination

	IPv4 Header	IPv6 Header	Description of the error
	Protocol Unreachable (code 2)	Type 4 code 1	Unrecognized Next Header type encountered
	Port Unreachable (code 3)	Type 1 code 4	Port unreachable
	Fragmentation Needed and DF was set (code 4)		No route to destination
	Source Route Failed(code 5)		No route to destination
	code 6	Type 1 code 0	No route to destination
	code 7	Type 1 code 0	No route to destination
	code 8	Type 1 code 0	No route to destination
	code 9	Type 1 code 1	
	code 10	Type 1 code 1	Communication with destination administratively prohibited
	code 11	Type 1 code 0	No route to destination
	code 12	Type 1 code 0	No route to destination
	code 13	Type 1 code 0	No route to destination
	code 14	Drop	
	code 15	Type 1 code 1	Communication with destination administratively prohibited
	Time Exceeded (type 11)	Type 3	Time Exceeded
Parameter Problem (Type 12)	Pointer indicates the error (code 0)	Type 4 code 0	Parameter Problem
	Missing a required option (code 1)	Drop	
	Bad Length (code 2)	Type 4 code 0	Erroneous header field encountered
	Echo request (Type 8)	Type 128	Echo Request
	Echo reply (Type 0)	Type 129	Echo Reply

Translating ICMP IPv6 Headers to ICMP IPv4 Headers

	IPv6 Header	IPv4 Header	Description of the error
Destination Unreachable (Type 1)	No route to destination (code 0)	code 1	No route to destination

	IPv6 Header	IPv4 Header	Description of the error
	Communication with destination prohibited (code 1)	code 10	Communication with destination prohibited
	Beyond scope 0f src address (code 2)	code 1	Beyond scope of src address
	Address unreachable (code 3)	code 1	Address unreachable
	Port unreachable (code 4)	code 3	Port unreachable
	Packet Too Big (Type 2)	Type 3 code 4	Fragmentation needed and Don't Fragment was Set
Time Exceeded (type 3)	Hop limit exceeded in transmit (code 0)	Type 11 code 0	Time to Live exceeded in Transit
	Fragment reassembly timeout (code 1)	Type 11 code 1	Fragment Reassembly Time Exceeded
Parameter Problem (Type 4)	Erroneous header field(code 0)	erroneous header field(code 0) Type 12 code 0	Pointer indicates the error
	Unrecognized nxt header field (code 1)	Type 3 code 2	Protocol Unreachable
	Unrecognized IPv6 option encountered (code 2)	drop	Communication with destination administratively prohibited



CHAPTER 6

External Logging

External logging configures the export and logging of the NAT table entries, private bindings that are associated with a particular global IP port address, and to use Netflow to export the NAT table entries.

- Bulk Port Allocation, on page 169
- Session logging, on page 170
- (Only NAT44) Syslog Logging, on page 170
- Reliable Log Transfer, on page 187
- Frequently Asked Questions (FAQs), on page 189

Bulk Port Allocation

The creation and deletion of NAT sessions lead to creation of logs. If logs of all such translations are stored, then a huge volume of data is created. This data is stored on a NetFlow or a Syslog collector. To reduce the volume of this data, a block of ports is allocated. If bulk port allocation is enabled, as soon as a subscriber creates the first session, a number of contiguous external ports are allocated. To indicate this allocation, a bulk allocation message is created in the log.

Note The bulk allocation message is created only during the first session. Rest of the sessions use one of the allocated ports. Hence no logs are created for them.

A bulk delete message is created in the log when the subscriber deletes all the sessions that are using the allocated ports.

Another pool of ports is allocated only if the number of simultaneous sessions is more than N where N is the size of the bulkk allocation. The size of the pool can be configured from the CLI.

Restrictions for Bulk Port Allocation

The restrictions for bulk port allocation are as follows:

- The value for the size of bulk allocation can be 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096. For optimum results, it is recommended that you set this size to half of the port limit.
- If the size of bulk allocation is changed, then all the current dynamic transactions will be deleted. Hence it is advisable to change the bulk port allocation size (only if necessary) during a maintenance window.

- The port numbers below the value of dynamic-port-range start value (which is 1024 by default), are not allocated in bulk.
- The algorithm that is used to allocate a public address to a user remains the same.
- When bulk allocation is enabled, session logging is not available.
- When bulk allocation is enabled, the translation record will not contain information about L4 protocol.
- Bulk port allocation features is not supported in NAT64 stateful application.

Session logging

In general, NAT translation entries contain information about private source IP, port and translated public IP and port. However, there could be cases when the destination IP address (public IP address) and port may also be needed. In such cases, session logging has to be enabled so that Netflow or Syslog translation records include these values as well.

(Only NAT44) Syslog Logging

Perform the following tasks to configure Syslog Logging for NAT table entries.

Restrictions for Syslog

The restrictions for syslog are as follows:

- Syslog is supported over UDP only.
- Syslog is supported in ASCII format only.
- You cannot log onto multiple collectors or relay agents.
- All the messages comply to RFC 3954 except for the timestamp format. Timestamp is represented in a simpler way as explained later in this section.
- Syslog shall be supported for DS-Lite and NAT444 as of now. Support for NAT64 is not yet available.

Syslog Message Format

In general, the syslog message is made up of header, structured data, and msg fields. However, in the CGv6 applications, the structured data is not used.

Header

The header fields shall be as per the RFC 5424. Fields shall be separated by ' ' (blank space) as per the RFC.

The header consists of the following fields:

Field	Description
Priority	• The priority value represents both the facility and severity. • Ensure that the severity code is set to Informational for all the messages at value 6.
Version	• This field denotes the version of the specification of the syslog protocol. • In CGv6 application, the version value is set to 1.
Timestamp	• This field is needed to trace the port usage. • The format is <year> <mon> <day> <hh:mm:ss>. • Ensure that the syslog collector converts the time to local time whenever needed. **Note** The timestamp is always reported in GMT/UTC irrespective of the time zone configured on the device.
Hostname	• This field is used to identify the device that sent the syslog message. • While configuring the syslog server, ensure that the host name does not exceed 31 characters. • The default value for the host name is '-'.
App name and PROC ID	These fields are not included. In ASCII format, '-' is included for these fields.
MSG ID	• This field identifies the type of the syslog message. • In the ASCII format, the values for NAT44 and DS Lite messages are NAT44 and DS LITE respectively.

Structured Data

It is not used.

MSG

This field consists of the information about the NAT44 or DS Lite events. In a single UDP packet, there could be one or more MSG fields each enclosed in [] brackets. The MSG field has many sub fields as it has a common structure across different records (for both NAT44 and DS Lite). Note, that, depending on the event, some of

the fields may not be applicable. For example, fields such as 'Original Source IPv6' address are not applicable for all NAT44 events. In such cases, the inapplicable fields will be replaced by '-'.

The syntax of the MSG part is as follows:

[EventName <L4> <Original Source IP> <Inside VRF Name> <Original Source IPv6> < Translated Source IP> <Original Port> <Translated First Source Port> <Translated Last Source Port> <Destination IP> <Destination Port>]

The descriptions of the fields in this format are as follows:

Field	Description
EventName	Select any one of the values for EventName from the following based on the event: • UserbasedA: User-based port assignment • SessionbasedA: Session-based port assignment • SessionbasedAD: Session-based port assignment with destination information Note: SessionbasedAD is used only if session logging is enabled. Also, session-logging and bulk port allocation are mutually exclusive. • UserbasedW: User-based port withdrawal • SessionbasedW: Session-based port withdrawal • SessionbasedWD: Session-based port withdrawal with destination information • Portblockrunout: Ports exhausted
L4	Specifies the identifier for the transport layer protocol. Select any one of the values for L4 from the following: • 1 for ICMP • 6 for TCP • 17 for UDP • 47 for GRE
Original Source IP	Specify the private IPv4 address.
Inside VRF Name	The Inside vrf is essential to identify the subscriber. Even though multiple subscribers connected to the router might have the same source IP, they might be placed in different VRFs. Hence the VRF name and the original source IP together helps to identify a subscriber. Ensure that the VRF name is not more than 32 characters in length.

Field	Description
Original Source IPv6	Specifies the IPv6 source address of the tunnel in case of DS Lite.
Translated Source IP	Specifies the public IPv4 address post translation.
Original Port	Specifies the source port number before translation. This is not applicable for the UserbasedA and UserbasedW events.
Translated First Source Port	Specifies the first source port after translation.
Translated Last Source Port	Specifies the last source port after translation. This is applicable only for the UserbasedA and UserbasedW events.
Destination IP	Specifies the destination IP recorded in the syslogs for the SessionbasedAD and SessionbasedWD events.
Destination Port	Specifies the destination port recorded in the syslogs for the SessionbasedAD and SessionbasedWD events.

Let us look at an example for NAT444 user-based UDP port translation mapping:

```
[UserbasedA - 10.0.0.1 Broadband - 100.1.1.1 - 2048 3071 - -]
```

The description for this example is as follows:

Value	Description
UserbasedA	Event Name
10.0.0.1	Original Source IP
Broadband	Inside VRF name
100.1.1.1	Translated Source IP
2048	Translated First Source Port
3071	Translated Last Source Port



Note The number of MSG fields in an UDP packet are determined by the following factors:

- The space available in the UDP packet depends on MTU.
- The translation events pertaining to MSG records in a given packet must have happened within a second (starting from the time at which the first event of that packet happened).

Netflow v9 Support

The NAT64 stateful, NAT44, and DS Lite features support Netflow for logging of the translation records.. The Netflow uses binary format and hence requires software to parse and present the translation records. However, for the same reason, Netflow requires lesser space than Syslog to preserve the logs.

Considerations

The considerations for NetFlow are as follows:

- NetFlow V9 is supported over UDP.
- You cannot log onto multiple collectors or relay agents.
- All the messages comply to RFC 3954.

NetFlow Record Format

As NetFlow V9 is based on templates, the record format contains a packet header and templates or data records based on templates.

Header

All the fields of the header follow the format prescribed in RFC 3954. The source ID field is composed of the IPv4 address of ServiceInfra interface (of the card) and specific CPU-core that is generating the record. The collector device can use the combination of the sourceIPv4Address field plus the Source ID field to associate an incoming NetFlow export packet with a unique instance of NetFlow on a particular device.

Templates for NAT44

The templates are defined and used for logging various NAT64 stateful, NAT44 and DS Lite events as follows. The templates may change in future software releases. Hence it is advised that the Netflow collector software is designed to understand the templates as distributed by the router and accordingly parse the records.

Options Templates

The translation entries consist of VRF IDs which might be incomprehensible to a user. To simplify this process, the CGv6 applications send the options templates along with the data templates.

Options template is a special type of data record that indicates the format of option data related to the process of NetFlow. The options data consist of the mapping between VRF Ids and VRF names. By parsing and using this data, the NetFlow collectors can modify the translation entries by adding VRF names instead of VRF IDs.

The value for the Template ID of options template is 1 where as the value of the Template ID for data template is 0. For more information on Options template, see RFC3954.

Events

The events and the corresponding template details are described in the following table:

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
Nat444 translation create event	256	Disabled	Disabled	ingressVRFID	234	4	ID of the Ingress VRF
				egressVRFID	235	4	ID of the Egress VRF
				sourceIPv4Address (pre-NAT)	8	4	Original Source IPv4 address
				postNATSourceIPv4 Address	225	4	Post NAT (outside) source IPV4 address
				sourceTransportPort (pre NAT)	7	2	Original source port
				sourceTransportPort	7	2	Post NAT (translated) source port
				protocolIdentifier	4	1	L4 protocol identifier

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
Nat444 session create event - session based (with destination)	271	Disabled	Enabled	ingressVRFID	234	4	ID of the Ingress VRF
				egressVRFID	235	4	ID of the Egress VRF
				sourceIPv4Address	8	4	Original source IPV4 address
				postNATSourceIPv4Address	225	4	Post NAT (outside) source IPV4 address
				sourceTransportPort	7	2	Original Source Port
				sourceTransportPort	7	2	Post NAT (translated) source port
				destinationIPv4Address	12	4	Destination IP address
				destinationTransportPort	11	2	Destination port
				protocolIdentifier	4	1	L4 protocol identifier

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
Nat444 translation create event - user based	265	Enabled	Disabled	ingressVRFID	234	4	ID of the Ingress VRF
				egressVRFID	235	4	ID of the Egress VRF
				sourceIPv4Address	8	4	Original source IPV4 address
				postNATSourceIPv4Address	225	4	Post NAT (outside) source IPV4 address
				postNATPortBlockStart	361	2	Start of Post NAT (translated) source port block
				postNATPortBlockEnd	362	2	End of Post NAT source port block

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
Nat444 translation delete event	257	Disabled		ingressVRFID	234	4	ID of the Ingress VRF
				sourceIPv4Address	8	4	Original source IPV4 address
				sourceTransportPort	7	2	Original source port
				protocolIdentifier	4	1	L4 protocol identifier
Nat444 session delete event - session based (with destination)	272	Disabled	Enabled	ingressVRFID	234	4	ID of the Ingress VRF
				sourceIPv4Address	8	4	Original source IPV4 address
				destinationIPv4Address	12	4	Destination IP address
				sourceTransportPort	7	2	Post NAT (translated) source port
				destinationTransportPort	11	2	Destination port
				protocolIdentifier	4	1	L4 protocol identifier

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
Nat444 translation delete event - user based	266	Disabled	Disabled	ingressVRFID	234	4	ID of the Ingress VRF
				sourceIPv4Address	8	4	Original source IPV4 address
				postNATPortBlockStart	361	2	Start of Post NAT (translated) source port block. Note this is not defined by IANA yet.
DS-Lite translation create event	267	Disabled	Disabled	ingressVRFID	234	4	ID of the Ingress VRF
				egressVRFID	235	4	ID of the Egress VRF

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
				Pre NAT Source IPv4 Address	8	4	Original source IPV4 address. This field is valid only when session logging is enabled. Else, it will be reported as 0
				Pre NAT Source IPv6 Address	27	16	IPv6 address of the B4 element (Tunnel source)
				postNATSourceIPv4Address	225	4	Post NAT (outside) source IPV4 address
				sourceTransportPort	7	2	Original source port
				sourceTransportPort	227	2	Post NAT (translated) source port

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
DS-Lite session create event - session based (with destination)	273	Disabled	Enabled	ingressVRFID	234	4	ID of the Ingress VRF
				egressVRFID	235	4	ID of the Egress VRF
				sourceIPv4Address	8	4	Original source IPV4 address
				sourceIPv6Address	27	16	IPv6 address of the B4 element (Tunnel source)
				postNATSourceIPv4Address	225	4	Post NAT (outside) source IPV4 address
				sourceTransportPort	7	2	Original source port
				sourceTransportPort	227	2	Post NAT (translated) source port
				destinationIPv4Address	12	4	Destination IP address
				destinationTransportPort	11	2	Destination port
				protocolIdentifier	4	1	L4 protocol identifier

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
DS-Lite translation create event - user based	269	Enabled	Disabled	ingressVRFID	234	4	ID of the Ingress VRF
				egressVRFID	235	4	ID of the Egress VRF
				sourceIPv4Address	8	4	Original source IPV4 address. This field is valid only when session logging is enabled. Else, it will be reported as 0
				sourceIPv6Address	27	16	IPv6 address of the B4 element (Tunnel source)

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
DS-Lite translation create event - user based				postNATSourceIPv4Address	225	4	Post NAT (outside) source IPV4 address
				postNATPortBlockStart	361	2	Start of Post NAT (translated) source port block
				postNATPortBlockEnd	362	2	End of Post NAT source port block
DS-Lite translation delete event	270	Disabled	Disabled	ingressVRFID	234	4	ID of the Ingress VRF
				sourceIPv4Address			Original source IPV4 address
				sourceIPv6Address			IPv6 address of the B4 element (Tunnel source)
				sourceTransportPort			Original source port
				protocolIdentifier			L4 protocol identifier

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
DS-Lite session delete event - session based (with destination)				ingressVRFID	234	4	ID of the Ingress VRF
				sourceIPv4Address	8	4	Original source IPV4 address
				sourceIPv6Address	27	16	IPv6 address of the B4 element (Tunnel source)
				sourceTransportPort	7	2	Original source port
				protocolIdentifier	4	1	L4 protocol identifier
DS-Lite translation delete event - user based	270	Disabled	Disabled	ingressVRFID	234	4	ingressVRFID
				sourceIPv4Address	8	4	Original source IPV4 address
				sourceIPv6Address	27	16	IPv6 address of the B4 element (Tunnel source)
				postNATPortBlockStart	361	2	Start of Post NAT (translated) source port block

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
NAT64 stateful translation create event	258	Disabled	Disabled	sourceIPv6Address	27	16	Source IPv6 address
				postNATSourceIPv4Address	225	4	Post NAT (outside) source IPV4 address
				sourceTransportPort	7	2	Original source port
				sourceTransportPort	227	2	Post NAT (translated) source port
				protocolIdentifier	4	1	L4 protocol identifier

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
NAT64 stateful session create event - session based (with destination)	260	Disabled	Enabled	sourceIPv6Address	27	16	Source IPv6 address (pre translation)
				postNATSourceIPv4Address	225	4	Post NAT (outside) source IPV4 address
				destinationIPv6Address	28	16	Destination IPv6 address (pre translation)
				Post translation Destination IP address	226	4	Destination IPv4 address (post translation)
				sourceTransportPort	7	2	Original source port
				sourceTransportPort	7	2	Post NAT (translated) source port
				destinationTransportPort	11	2	Destination port
				protocolIdentifier	4	1	L4 protocol identifier

Event	Template ID	Bulk Port Allocation	Destination/Session Logging	Field Name	IANA IPFIX ID	Size in bytes	Description
NAT64 translation delete event	259	Disabled	Disabled	sourceIPv6Address	27	16	IPv6 address of the B4 element (Tunnel source)
				sourceTransportPort	7	2	Original source port
				protocolIdentifier	4	1	L4 protocol identifier
NAT64 stateful session delete event - session based (with destination)	261	Disabled	Enabled	sourceIPv6Address	27	16	IPv6 address of the B4 element (Tunnel source)
				destinationIPv6Address	28	16	Destination IPv6 address (pre translation)
				sourceTransportPort	7	2	Original source port
				destinationTransportPort	11	2	Destination port
				protocolIdentifier	4	1	L4 protocol identifier

Reliable Log Transfer

The VSM Line Cards support CGN application with NAT 44 feature. NAT 44 feature performs IPv4 NAT operations. The CGN applications based on server configuration generates NetFlow and Syslog records that contain critical NAT information. If an external NetFlow server is configured for storage and retrieval of NAT information, the information is transferred to the server using UDP connection.

The CGN application VM running on VSM also supports transfer of records to external NetFlow servers using TCP. This provides reliable log transfer. Use the *protocol* command to define the protocol to be used for the transfer.

Limitations

- Netflow and syslog records may be lost during any reload of VSM or restart of IOS XR VM or process.
- A maximum of two NetFlow or Syslog servers are supported.
- Server IP address along with the port number defines one server.
- Same TCP server can be configured on any number of VRFs and is considered as one TCP server.
- Loss of TCP connection may result in loss of NetFlow records.
- Bulk port allocation needs to be enabled with a minimum allocation value of 256 to prevent loss of NAT 44 records, when Syslog and Netflow records are sent using TCP protocol.

Reliable Log Transfer Configuration: Examples

Configure NetFlow records' transport using TCP protocol when the TCP server is in the default global VRF instance:

```
service cgn cgn1
 service-location preferred-active 0/3/CPU0
 service-type nat44 nat1
  inside-vrf insidevrf1
   map outside-vrf outsidevrf1 outsideServiceApp ServiceApp2 address-pool 100.1.1.0/24
   external-logging netflow version 9
    server
     address 45.1.1.1 port 65535
      protocol tcp
```

Configure NetFlow records' transport using TCP protocol when the TCP server is in an non-default VRF instance:

```
service cgn cgn1
 service-location preferred-active 0/3/CPU0
 service-type nat44 nat1
  inside-vrf insidevrf1
   map outside-vrf outsidevrf1 outsideServiceApp ServiceApp2 address-pool 100.1.1.0/24
   external-logging netflow version 9
    server
     address 55.1.1.1 port 4500
      protocol tcp vrf netflow-srv-vrf
```

Disabling NetFlow records' transport:

```
service cgn cgn1
service-type nat44 nat1
 inside-vrf insidevrf1
  no external-logging netflow version 9
```

Configure Syslog records' transport using TCP protocol when the TCP server is in the default global VRF instance:

```
service cgn cgn1
 service-location preferred-active 0/3/CPU0
 service-type nat44 nat1
  inside-vrf insidevrf1
   map outside-vrf outsidevrf1 outsideServiceApp ServiceApp2 address-pool 100.1.1.0/24
   external-logging syslog
    server
     address 45.1.1.1 port 3500
      protocol tcp
```

Configure Syslog records' transport using TCP protocol when the TCP server is in an non-default VRF instance:

```
service cgn cgn1
 service-location preferred-active 0/3/CPU0
 service-type nat44 nat1
  inside-vrf insidevrf1
   map outside-vrf outsidevrf1 outsideServiceApp ServiceApp2 address-pool 100.1.1.0/24
   external-logging syslog
    server
     address 55.1.1.1 port 4500
      protocol tcp vrf syslog-srv-vrf
```

Disabling Syslog records' transport:

```
service cgn cgn1
 service-type nat44 nat1
  inside-vrf insidevrf1
   no external-logging syslog
```

Frequently Asked Questions (FAQs)

This section provides answers to the following frequently asked questions on external logging.

Q: How to trace a subscriber by using the NAT logs?

A: In order to trace a subscriber, you should know the public source IP address (post NAT source address), post NAT source port, protocol, and the time of usage. With these parameters, the steps to trace a subscriber are as follows:

1. Search for the create event that has the matching public IP address, post NAT Source IP address (postNATSourceIPv4Address) and protocol, egress VRF ID/Name and the time of the usage. Ensure that the time of the create-event is the same or earlier than the time of usage reported. You may not find the protocol entry or the exact post NAT source port in the logs if bulk allocation is enabled. In such cases, find the create-event whose **Post NAT Port Block Start** and **Post NAT Port Block End** values include the post NAT source port. The **Pre NAT source IP address** along with the corresponding **ingress VRF ID/Name** will identify the subscriber.
2. The corresponding delete record may be found optionally to confirm that the subscriber was using the specified public IP and port during the time of the reported usage.

Q: The Netflow records provide VRF IDs for ingress and egress VRFs. How will I know the VRF names?

A: The following are the two ways to find the VRF name from the VRF ID.

1. Use the command show rsi vrf-id <vrf-id> on the Router console to find VRF-ID to VRF-NAME associations.

2. The CGv6 applications periodically send out option templates containing the VRF-ID to VRF-NAME mapping. The Netflow collector software presents the information with VRF-Names rather than VRF IDs.

Q: Does the time format in Syslog or Netflow account for Day light saving?

A: The Syslog and Netflow formats report time corresponding to GMT/UTC. The Netflow header contains the time in seconds that elapsed since EPOCH whereas the Syslog header contains time in human readable formats. In both cases, the day light saving is not accounted. The Netflow/Syslog collectors have to make that adjustments if needed.

Q: Since the Netflow and Syslog use UDP, how can we know if a packet containing translation record was lost?

A: The Netflow header contains a field called Sequence Number. This number is indicates the count of the packet coming from each Source ID. The Netflow collector traces the Seqence Number pertaining to each unique Source ID. The sequence numbers should be increased by one for each packet sent out by the Source. If the collector ever receives two successive packets with the same Source ID, but with a Sequence number difference of more than 1, it indicate a packet loss. However, currently, no such mechanism exists for Syslog.

Q: What is the use of session-logging?

A: Session logging includes destination IP and port number as well. Though this information is not directly useful in tracing the subscriber, in some cases, this information may be useful or may be mandated by the legal authorities. There are cases where, legal authorities may not have the post NAT source 'port', however may know the destination IP address (and optionally destination port, such as IP address and port of an e-mail server). In the absence of post NAT source port information, a list of subscribers who used the specified public IP during that time may have to be pruned further based on the destination IP and port information.

Q: How does the bulk port allocation reduce data volume of translation logs?

A: With bulk port allocation, subscribers are allocated a range of contiguous ports on a public IP. Quite often, a subscriber will need more ports than just one. Especially AJAX based web pages and other web applications simultaneously open several ports. In such cases, pre-allocated ports are used and only one log entry is made that specifies the range of ports allocated to the user. Hence, bulk port allocation significantly reduces log data volume and hence the demand on storage space needed for the translation logs.

Q: What else can be done to reduce log data volume?

A: Predefined NAT is an option that can be used to eliminate the logging altogether. The Predefined NAT translates private IP address to public IP address and a certain port range by using an algorithm. Hence there is no need to keep track of NAT entries.