Configuring Layer 3 and Layer 4 Features

This chapter provides information about configuring Layer 3 and Layer 4 features on the Cisco 7600 Series Ethernet Services Plus (ES+) line card family (ES+, ES+T, ES+XT, ES+XC). It includes the following topics:

- Layer 3 and Layer 4 Security ACL on Service Instance, page 13-1
- Inline Video Monitoring on the Cisco 7600 Router, page 13-5
- IP Tunneling - IPv6 Rapid Deployment, page 13-31
  - VRF aware IPv6 Rapid Deployment (6RD) tunnels, page 13-38
- VRF aware IPv4 Tunnels over IPv4 Transport, page 13-62
- IPv6 over IPv4-GRE Tunnels, page 13-78
- IPv6 Policy Based Routing, page 13-92


Note: The information provided in this chapter is applicable to all the ES+ line card family unless specified otherwise.

Layer 3 and Layer 4 Security ACL on Service Instance

An ACL consists of a series of statements called ACL entries that define the network traffic profile. Each entry permits or denies network traffic (inbound and outbound) to the parts of your network specified in the entry. Each entry also contains a filter element that is based on criteria such as the source address, the destination address, the protocol, and protocol-specific parameters such as ports and so on.

The Layer 3 and Layer 4 ACLs on Service Instance feature permits you to configure ACLs under an Ethernet Virtual Circuit (EVC) on the Cisco 7600 Series ES+ line cards. Cisco IOS Release 15.1(1)S supports EVC port-channels.

Restrictions and Usage Guidelines

When configuring the Layer 3 and Layer 4 Security ACL on Service Instance feature on Cisco 7600 Series ES+ line cards, follow these restrictions and usage guidelines:
- 8000 unique ACLs are supported per NP
- 8000 ACEs are supported per ACL with only single ACL present
- 8000 EVCs are supported per NP
- If TCAM is full, filtering is not supported
- IPv6 ACLs are not supported
- Operators for Layer 4 attributes are not supported
- `time-range`, `dynamic range`, and `acl log` are not supported
- Layer 2 and Layer 3 ACLs cannot coexist on the same service instance
- 8000 access control entries (ACEs) per ACL on EVC
- The number of uniquely defined ACLs on the chassis is not affected by support on service instances
- ACL configuration with ACEs that contain type of service (ToS) configuration is not supported, but differentiated services code point (DSCP) is supported
- IP options are not supported.

### Configuring on a Service Instance

#### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `interface gigabitethernet slot/port` or `interface tengigabitethernet slot/port`
4. `[no] ip address`
5. `service instance id ethernet [service-name]`
6. `ip access-group {access-list-name | access-list-number} {in | out}`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td><code>Router# enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
</tbody>
</table>
## Layer 3 and Layer 4 Security ACL on Service Instance

### Configuring on a Port-Channel

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `interface port-channel number`
4. `[no] ip address`
5. `service instance id ethernet [service-name]`
6. `ip access-group {access-list-name | access-list-number} {in | out}`

### Command

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>interface gigabitethernet slot/port</code></td>
<td>Specifies the Gigabit Ethernet or the Ten Gigabit Ethernet interface to configure, where:</td>
</tr>
<tr>
<td></td>
<td>or <code>interface tengigabitethernet slot/port</code></td>
<td>• <code>slot/port</code>—Specifies the location of the interface.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config)# interface gigabitethernet 4/1</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>[no] ip address</code></td>
<td>Assigns an IP address and subnet mask to the EtherChannel.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-if)# no ip address</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>service instance id ethernet [service-name]</code></td>
<td>Creates a service instance (an instantiation of an EVC) on an interface and sets the device into the config-if-srv submode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-if)# service instance 101 ethernet</code></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>`ip access-group {access-list-name</td>
<td>access-list-number} {in</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-if-srv)# ip access-group 101 out</code></td>
<td></td>
</tr>
</tbody>
</table>
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> interface port-channel number</td>
<td>Creates the port-channel interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# interface port-channel 12</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> [no] ip address</td>
<td>Assigns an IP address and subnet mask to the EtherChannel.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-if)# no ip address</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> service instance id ethernet [service-name]</td>
<td>Creates a service instance (an instantiation of an EVC) on an interface and sets the device into the config-if-srv submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-if)# service instance 101 ethernet</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> ip access-group {access-list-name</td>
<td>access-list-number} {in</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-if)# ip access-group 101 out</td>
<td></td>
</tr>
</tbody>
</table>

### Examples

In this example, the Layer 3 access control list below is applied under the EVC and a port-channel on a Cisco ES+ line card.

```
ip access-list extended 13acl
    permit ip 1.1.1.1 255.255.255.255 any
    permit ip 2.2.2.2 255.255.255.255 any
```

**Example:**

```bash
Router# enable
Router# configure terminal
Router(config)# interface GigabitEthernet 3/1
Router(config-if)# no ip address
Router(config-if)# service instance 1 ethernet
Router(config-if-srv)# ip access-group 13acl in/out
```

**Example:**

```
Router# enable
Router# configure terminal
```
In this example, the Layer 4 access control list below is applied under the EVC and a port-channel on a Cisco ES+ line card.

```
ip access-list extended l4acl
   permit tcp host 1.1.1.1 eq 30 any
```

```
Router# enable
Router# configure terminal
Router(config)# interface GigabitEthernet 3/1
Router(config-if)# no ip address
Router(config-if)# service instance 1 ethernet
Router(config-if-srv)# ip access-group l4acl in/out
```

### Verification

Use the following commands to verify operation.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# show ethernet service evc [id evc-id</td>
<td>interface interface-id] [detail]</td>
</tr>
<tr>
<td>Router# show ethernet service instance [id instance-id interface interface-id</td>
<td>interface interface-id] [detail]</td>
</tr>
</tbody>
</table>

### Inline Video Monitoring on the Cisco 7600 Router

IP video is highly sensitive to delay and packet loss. It is estimated that nearly twenty percent of the access lines are of marginal quality, and a three millisecond (ms) packet loss results in a 500-1000 ms video degradation visible to the subscriber. A data loss in a multicast video stream in the core network affects multiple access and aggregation networks, and thousands of subscribers viewing the stream.

This feature provides the functionality for inline video monitoring. Using the inline video monitoring, you can monitor the video inline in the router without using a video probe. A Video probe is an external device used for video monitoring. Video Monitoring feature enables you to monitor the video data flow in a network. These features are included in Inline Video Monitoring.
Media Delivery Index

The Media Delivery Index (MDI) metric provides a relative indicator of the required buffer depths at the consumer node due to packet jitter. It also gives an indication of the lost packets. MDI provides the Delay Factor (DF) and the Media Loss Rate (MLR) for the video flow. DF is the maximum difference between the arrival of a packet and the drain of the packet. MLR is the number of media packets lost over a certain time interval. Media Discontinuity Count (MDC) is a measure of the number of times discontinuity events occurred resulting in MLR. MDC metric is a Cisco proprietary standard.

**Note**

In case of major data loss, the reported MDC & MLR values are capped to 65535 for an interval and reset to zero from the next interval.

**Note**

The maximum value for DF is capped to 1000 ms.

Support for IP Delay Variation for 7600 Inline Video Monitoring

Previously, for inline video monitoring, the jitter buffer required by the end devices was calculated using the delay factor (DF) algorithm defined in RFC 4445. This algorithm was effective for constant bit rate (CBR) flows where the flow rate was consistent and helped calculate the DF accurately. However, in a variable bit rate (VBR) flow or inconsistent flow rate, the calculated DF was inaccurate, hence not very helpful to the service provider.

Effective from release 15.1(1)S, video monitoring on the Cisco 7600 Series Routers supports DF computation as IP Delay Variation (IPDV). The IPDV algorithm is independent of configured packet rate and is useful for a service provider to calculate accurate jitter buffer for VBR flows. You can configure either RFC 4445 or IPDV algorithm on a flow to calculate the DF. To configure the delay factor using either of these algorithms, use `df algo_type` command.

**Note**

DF computed using RFC4445 algorithm includes the inter-packet gap and hence it is never zero. But IPDV does not include the inter-packet gap and the computed DF can be zero.

These are the characteristics of IPDV configuration:

- IPDV or MDI-DF on a per class basis with-in a policy-map is supported.
- IPDV and MDI-DF can coexist within the same policy-map.
Internet Protocol-Constant Bit Rate (IP-CBR)

The Internet Protocol-Constant Bit Rate (IP-CBR) metric provides the Media Rate Variation (MRV) and Delay Factor (DF). MRV is used on CBR flows to isolate the variations in the data transport due to packet loss. The MRV metric indicates the percentage rate of variation of media from the expected metrics calculated rate. MRV is calculated based on the expected bit rate provided by the user and the actual bit rate. Delay factor is the measured difference between the arrival of a packet and the drain of the packet.

Advantages of using IPDV to calculate DF:
- IPDV cannot co-exist with MDI-DF under the same class-map.
- IPDV and MDI-DF can be configured under different class-maps under the same policy-map.

Note
The maximum value for DF is restricted to 1000 milliseconds.

Support MPLS Encapsulation for 7600 Inline Video Monitoring

Inline video monitoring feature monitors MPLS encapsulated video packets on MPLS enabled interfaces. Effective from Cisco IOS release 15.1(1)S, inline video monitoring is also supported for these MPLS scenarios:
- Tag to Tag: 7600 router configured as Label Switch Router (LSR) to switch MPLS packets.
- Tag to IP: 7600 router configured as Label Edge Router (LER) to remove the last MPLS tag.
- IP to Tag: 7600 router configured as LER to add the first MPLS tag.

The following MPLS packet formats are supported for inline video monitoring:
- L3VPN packet formats: 0x8847, MPLS Labels, IP header, UDP header, and MPEG. (ignore acronyms)
- L2VPN and VPLS packet formats: Router MAC, 0x8847, MPLS Labels, control word, VLAN Tags, CE MAC, IP, UDP, and MPEG.

Configurable MPEG Video PIDs for Inline Video Monitoring

Until Cisco IOS release 15.0(1)S, inline video monitoring learned the first five unique Program Identifiers (PIDs) in an MPEG flow for video, audio, or caption data PIDs. However, monitoring PIDs for audio or caption data is not a priority for a customer implementing inline video monitoring. Effective from Cisco IOS release 15.1(1)S, video monitoring provides support to configure the PIDs to monitor. This enables a user to configure only the video PIDs in an MPEG flow on priority. The PIDs to monitor are configured within the `monitor metric mdi` command mode using the `monitor pids pid_value` command. You can configure a maximum of five PIDs using this command. The PID value can range from hexadecimal value 2 to 1FF.

Note
This feature is supported on flows monitored for MDI metrics.
RTP Metrics support for 7600 Inline Video Monitoring

Real-time Transport Protocol (RTP) provides protocol level support for detecting packet loss and jitter in a network. Packet loss is detected using the 16 bit sequence numbers in the packet header. These numbers provide an accurate measurement of number of packets lost and delayed during transmission. The timestamp information in the RTP packet header is used for calculating jitter in a network data stream. Effective from Release 15.1(2)S, inline video monitoring supports monitoring packet loss and jitter metrics for RTP flows in addition to IP-CBR and MPEG flow.

RTP metrics is enabled on a per class-map basis on the Cisco 7600 series routers. A new RTP flow is created for each RTP Synchronization Source (SSRC) detected in the RTP session matching the class-map classification criteria. Since RTP sessions are dynamically negotiated, they must be validated before learning the RTP flow for monitoring. A RTP header does not contain protocol specific information to identify it as an RTP packets. Currently, these checks are performed to ensure that a particular RTP packet is valid:

- The RTP version number should be two.
- The payload type should be known and not equal to SR (Sender Report 200) or RR (Receiver Report 201).
- When the SSRC identifier is received for the first time, the data packets carrying the identifier are considered invalid until a number of data packets with consecutive sequence numbers are received.
- The SSRC value should not be zero.

Note

RTP SSRC is a part of flow key along with existing five flow tuples.

RTP Metrics

Apart from the packet loss and jitter metrics, an RTP flow contains additional metrics that provide information about the RTP traffic. Table 13-1 lists the metrics exported and displayed for an RTP flow.

Table 13-1 RTP Reported Metrics

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Description</th>
<th>Cumulative/ Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>total_pkts</td>
<td>Total number of packets monitored for the interval.</td>
<td>Interval + Cumulative</td>
</tr>
<tr>
<td>expected_pkts</td>
<td>Total number of packets expected in an interval based on the minimum and maximum sequence numbers.</td>
<td>Interval + Cumulative</td>
</tr>
<tr>
<td>lost_pkts</td>
<td>Total number of packets lost in an interval. It is the difference between the expected (expected_pkts) and the actual packets (actual_pkts).</td>
<td>Interval + Cumulative</td>
</tr>
<tr>
<td>jitter</td>
<td>Jitter reported for an interval.</td>
<td>Interval</td>
</tr>
<tr>
<td>max_jitter</td>
<td>Maximum jitter observed in the interval.</td>
<td>Interval</td>
</tr>
<tr>
<td>loss_intervals</td>
<td>Number of loss intervals(^1). A loss interval is an interval when the consecutive RTP packets were lost.</td>
<td>Interval</td>
</tr>
</tbody>
</table>
Chapter 13      Configuring Layer 3 and Layer 4 Features

Support Switch-Port Interfaces for 7600 Inline Video Monitoring

Effective from Release 15.1(2)S, inline video monitoring feature supports video traffic monitoring on layer 2 and layer 3 interfaces. These layer 2 switch-port interfaces are supported:

- Trunk interface: When you configure a switch-port mode as trunk, multiple VLANs can be switched on the interface.
- Access interface: When you configure a switch-port mode as access, a single VLAN can be switched on the interface.
- Dot1q tunnel: When you configure a switch-port mode on the router as trunk and on the peer as non-trunk or vice-versa.

Note: Apart from the five tuple keys, inner and outer VLAN ids can be used as a key to differentiate flows.

Support PPPoE Encapsulation for 7600 Inline Video Monitoring

Point-to-Point Protocol over Ethernet (PPPoE) is a network protocol for encapsulating Point-to-Point Protocol (PPP) frames inside the ethernet frames. This protocol is used for Digital Subscriber Line (DSL) services where a user in a metro ethernet network, connects to the DSL modem. Effective from Release 15.1(2)S, inline video monitoring supports to monitor video traffic from a PPPoE network. Packets with ether type as 0x8864 are considered as the PPPoE packets and included for video monitoring.

These PPPoE encapsulated packet formats are supported:

- PPPoE packets
  - Eth + VLAN + PPPoE + IP

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Description</th>
<th>Cumulative/Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>num_resync</td>
<td>Total number of sequence number re-synchronizations performed in an interval.</td>
<td>Interval + Cumulative</td>
</tr>
<tr>
<td>late_pkts</td>
<td>Total number of packets received outside the sliding window defined by maximum reorder (max_reorder) and dropout (max_dropout) parameters.</td>
<td>Interval</td>
</tr>
<tr>
<td>reord_pkts</td>
<td>Total number of reordered packets received in a interval.</td>
<td>Interval</td>
</tr>
<tr>
<td>lost_fraction</td>
<td>The number of packets lost divided by the number of packets expected. Displayed in percent.</td>
<td>Interval</td>
</tr>
<tr>
<td>avg loss duration</td>
<td>The number of packets lost (lost_pkts) divided by the number of loss intervals (loss_intervals).</td>
<td>Interval</td>
</tr>
<tr>
<td>valid packets</td>
<td>Difference between the number of packets received and the number of reordered and late packets.</td>
<td>Interval</td>
</tr>
</tbody>
</table>

1. For loss interval calculations any late or reordered packets should also be treated as lost.
Inline Video Monitoring on the Cisco 7600 Router

Chapter 13      Configuring Layer 3 and Layer 4 Features

- L2VPN
  - Eth + MPLS + Eth + VLAN + PPPoE + IP
  - Eth + VLAN + MPLS + Eth + VLAN + PPPoE + IP

Note: Video monitoring for PPPoE encapsulated packets is not supported on a node where the session terminates.

Inline Video Monitoring Support of MDI Metrics for RTP Encapsulated Flows

Effective with Release 15.1(3)S, inline video monitoring supports MDI metrics calculation for MPEG2-Transport Stream (TS) flows encapsulated in RTP (RFC3550) headers. The MDI metric (RFC4445) provides information about the buffer required at the consumer node for packet jitter (DF) and an estimate of the packet loss during the data transmission (MLR/MDC).

Note: Currently, you can monitor either the MDI or RTP at a time for data flow, not both together.

Inline Video Monitoring Support for Availability Metrics

Effective with Release 15.1(3), inline video monitoring provides an availability metrics named transport-availability, which indicates the availability of a transport stream for a specific period of time. Inline video monitoring computes transport-availability and error-seconds based on either MDI (RFC 4445) or RTP (RFC 3550) metrics. These metrics provide network operators additional troubleshooting information and the option to measure per video flow performance against the defined Service Level Agreements (SLA).

Note: Before Cisco IOS Release 15.1(3), inline video monitoring provided metrics such as MLR and DF for MDI traffic, and jitter and loss-fraction for RTP traffic. To understand these metrics, a user should have an understanding of technology and standards.

Transport-availability is calculated as the percentage of time a transport stream is available over a measured time interval, and the error-seconds (downtime) is the time interval for which the transport stream in not available for data transmission. The transport-availability is calculated as:

\[
\text{Transport-availability} = \frac{\text{Interval duration} - \text{Error-seconds}}{\text{Interval duration}}
\]

Note: Two new react-types, \textit{transport-availability} and \textit{error-seconds}, are introduced in the \texttt{react} command to help configure alarms based on the keyword values.

Note: Packet drop occurs during the error-seconds interval.

Note: Only the packet loss is considered for calculating error-seconds; jitter is not considered for error-seconds calculation.
Inline Video Monitoring Support for Uncompressed Video

Effective with Release 15.1(3)S, inline video monitoring supports monitoring of uncompressed video, such as Serial Data Interface (SDI) and High Definition-SDI (HD-SDI). RTP loss metrics are not frequency dependent, and jitter calculation involves frequency. Hence, with the existing default 90kHz frequency, jitter calculation for higher frequencies might display incorrect results. To monitor uncompressed videos, three new RTP clock frequencies: 148.5MHz, 148.5/100, and 27MHz are supported apart from existing support for 90kHz. You can configure the RTP clock frequency using the `clock-rate` command. This command allows you to map a dynamic Payload Type (PT) value to the corresponding frequency for each class-map in the RTP header. Based on the PT value in the RTP header for a flow, a corresponding frequency is mapped for jitter calculations. For the un-mapped PTs, default frequency of 90kHz is used.

- **Note**
  You can disable jitter calculation for unsupported frequencies. The jitter value for unsupported frequencies is reported as 0.

Restrictions for Inline Video Monitoring

The following restrictions apply to the inline Video Monitoring feature:

- Video Monitoring is supported only on ES+ line cards.
- The supported supervisor engines are Sup720 and RSP720 (1 gigabits and 10 gigabits).
- Up to 1000 video monitoring flows per Line Card and up to 8000 flows per router are supported for inline video monitoring.
- Only IPv4 ACLs are supported.
- The video traffic is not monitored up to first two intervals after the flow is learnt.
- After the LC flow traffic stops and is timed out using the configured timeout value under class-map, some of the system resources are released only after 25 minutes. The learn-delete process may result in delay in monitoring the flows because the system resources are not released immediately.
- In case of video monitoring on EVC, monitoring is performed for learnt unicast and multicast traffic only. Traffic with unknown unicast destination MAC is not monitored.
- MDI:DF, MDI:MLR, MDI:MDC, IP-CBR:DF, and IP-CBR:MRV metrics are supported for CBR flows. For VBR flows, only MDI:MLR and MDI:MDC are supported.
- MDI:DF, MDI:MLR, and MDI:MDC are supported only for MPEG-2 and MPEG-4 transport streams. Both the single program transport streams (SPTS) and multi-program transport streams (MPTS) are supported.
- Only a flat performance-traffic policy type can be configured in each direction. Hierarchical policies are not supported for Video Monitoring in the `performance-traffic` typed policy.
- Video Monitoring is an independent feature and can co-exist with QoS. Though QoS and performance-traffic are policy-map based, both can be applied to the same interface in the same direction to function independently.
- A maximum of five PIDs can be configured for monitoring.
- Only the configured PIDs are monitored. For example, if only one PID is configured, no other new PIDs are monitored.
- These reserved PIDs are not monitored:
– 0x0000: Reserved for Program Association Table (PAT).
– 0x0001: Reserved for Conditional Access Table (CAT).
– 0x0010: Reserved for Network Information Table (NIT).
– 0x1FFF: Reserved for Null Packets.

• Duplicate PID values cannot be configured.
• Layer 3 VPN (L3VPN) and Layer 2 VPN (L2VPN)/Virtual Private LAN Services (VPLS) MPLS encapsulated packet format are supported.
• Flow from a CE MAC and IP HDR magic pattern is not supported.
• MPLS labels and EXP values are not supported as part of the flow key. If two different customers using different MPLS labels but same IP address and UDP ports are on the same target, both are mapped to the same video monitoring flow.
• MDI-DF and IPDV cannot be configured on the same class-map.
• RTP metric cannot co-exist with MDI or IP-CBR in the same class-map.
• Clock rate support is limited to 90Khz. Jitter metric computation accuracy is not guaranteed if the clock rate for packets is not 90Khz.
• Performance-type policy-map is supported on switch-port trunk mode, access mode, and Dot1q tunnel mode.
• PPPoE control packets are not monitored.
• These flows are not supported:
  – Fragmented IPv4 packets
  – Tunneled GRE, mGRE, L2TPv3, or multicast VPN
• IPv6 and tunneled IPv6
• MPEG transport streams where TS header is encrypted
• The value of error-seconds metrics ranges from 0 to 1000.
• Transport-availability and error-seconds metrics are not calculated for IP-CBR flows.
• Static payload types 1 to 95 can only be mapped to the frequency option disable.

Supported Interfaces

Video Monitoring is supported on the routed main interface, subinterfaces, switchports, and EVCs in release 15.0(01)S.

Table 13-2 lists the inline video monitoring interface support for each release:

<table>
<thead>
<tr>
<th>Cisco IOS Release</th>
<th>Interfaces Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33) ZI</td>
<td>Main-interface, Sub-interface.</td>
</tr>
<tr>
<td>15.0(1)</td>
<td>Main-interface, Sub-interface, EVCs.</td>
</tr>
<tr>
<td>15.1(1)</td>
<td>Main-interface, Sub-interface, EVCs.</td>
</tr>
<tr>
<td>15.1(2)</td>
<td>Main-interface, Sub-interface, EVCs, L2 switch-port interface.</td>
</tr>
</tbody>
</table>


Ingress and Egress Interfaces

Video Monitoring can be configured on both ingress and egress interfaces. The following types of monitoring is allowed on these interfaces:

- Ingress only monitoring
- Egress only monitoring
- Ingress and egress for the different flows on different ports.
- Ingress and egress for the same flow

Monitored Video Flows

Video Monitoring feature supports only UDP traffic in release 15.0(01)S. The following flows are monitored:

- IP+UDP
- Single program transport streams (SPTS) and multi-program transport streams (MPTS)
- MPEG-2 and MPEG-4
- MPLS+IP+UDP
- IP+UDP+RTP

Alerts and Event Notifications

Alerts and notifications enable you to track the performance in a system. The flow of video can be tracked and managed using alerts and event notification. Computed metric values are used to generate alerts and event notifications.

Media Stop Events

Media Stop Event is triggered when no packets are received for at least eight seconds on a valid flow for a configured interval. The reason for MSE can be:

- Media Server failure
- Upstream network failure
- Genuine flow ending.

MSE interval causes invalidation of metrics data for up to two subsequent intervals. Metrics from these invalidated intervals do not trigger any traps or reacts.

Threshold Crossing Alerts

Router reports the metric values at the end of the monitoring interval. The computed values are compared with the configured threshold react range and an alarm is triggered if the computed value is not within the configured range. The router relays the alerts to the management station through a SNMP trap.
notification. The alerts can be immediate or average. An immediate alert is triggered at the end of monitoring interval if the metric value crosses the configured range. An average alert is sent based on the average value, which is computed based on the last \( n \) monitored intervals.

**Note**
If two alerts are asserted for a same interval, the alert with lower profile-id is asserted. The alert profiles with lower profile-id have higher priority.

### Flow Monitoring and Metric Computation

This section describes how to configure the Video Monitoring feature and report the metrics.

#### Provisioning the Metric

Provisioning the metric involves creating a policy map, defining the filtering criteria, and applying the policy map on an interface. A new policy map type `performance-traffic` is used for Video Monitoring. The policy map contains a list of actions for the flow monitoring.

**Note**
The maximum number of class maps supported in a performance-traffic policy map is 50. The maximum number of policy maps (including QoS and typed policy maps) supported on a router is 1023.

Follow these steps to configure video monitoring on an interface:

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `access-list access-list-number permit ip [host] source destination`
4. `class-map [match-any] class-map-name`
5. `match access-group access-group-name | access-group-number`
6. `exit`
7. `policy-map type performance-traffic policy-map-name`
8. `class class-map-name`
9. `monitor parameters`
10. `df rfc4445 \ ipdv`
11. `interval duration n-secs`
12. `timeout n-interval`
13. `history n-interval`
14. `exit`
15. `monitor metric [mdi \ ip-cbr1rtp]`
16. (optional) `clock-rate dynamic-pt frequency`
17. (optional) `monitor pids pid1 [pid2] [pid3] [pid4] [pid5]`
18. `rate [media \ layer3] [packet n-ppps [pps] | n {bps \ kbps \ mbps \ gbps}]`
### Chapter 13 Configuring Layer 3 and Layer 4 Features

#### Inline Video Monitoring on the Cisco 7600 Router

19. `packet {size media n-bytes | media in-layer3 n-packets}`

20. `react profile id-value {mdi-df | mdi-mdc | mdi-mlr | ip-cbr-mrv | ip-cbr-df | media-stop | rtp-lost-fraction | rtp-jitter | rtp-max-jitter | rtp-lost-pkts | transport-availability | error-seconds}`

21. `threshold {range range-value1 range-value2} | {[gt|ge|lt|le] value3} | {type [immediate | average value4]}`

22. `action {syslog | snmp}`

23. `alarm severity {none | informational | notification | warning | error | critical | alert | emergency}`

24. `alarm type discrete`

25. `description character string`

26. `interface type number`

27. (optional) `service instance instance-number ethernet`

28. `service-policy type performance-traffic {input | output} {policy-map name}`

29. `exit`

### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>router&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong> access-list access-list-number permit ip [host] source destination</td>
<td>Identifies the flow to be monitored.</td>
</tr>
<tr>
<td>Example:</td>
<td>router(config)# access-list 101 permit ip host 10.10.2.20 any</td>
</tr>
</tbody>
</table>

**Note**

Classification based on IP address, precedence, and DSCP values is supported for extended access-list.

**Note**

The *deny* option for *access-list* command is not supported. If *deny* option is configured under the *access-list* command and the class-map configured with the *deny* condition is part of the performance-traffic policy map, the video traffic is not monitored.
### Inline Video Monitoring on the Cisco 7600 Router

**Chapter 13  Configuring Layer 3 and Layer 4 Features**

**Step 4**

```
class-map match-any class-map-name
```

Example: router(config)# class-map match-any video-class

**Purpose**

Defines a class map.

In this example, a class-map named `video class` is defined with match criteria `match any`. The packets must meet any of the match criteria in the class map `video-class`.

**Step 5**

```
match access-group access-group-name
```

Example: router(config-cmap)# match access-group 101

**Purpose**

Defines the access-group. Only IPv4 acls are supported for Video Monitoring.

**Step 6**

```
exit
```

Example: router(config-cmap)# exit

**Purpose**

Exits class-map configuration mode.

**Step 7**

```
policy-map type performance-traffic policy-map name
```

Example: router(config)# policy-map type performance-traffic video-monitor

**Purpose**

Creates a `performance-traffic` type policy map and enters the policy map configuration mode.

In this example, the type of the `policy-map` is `performance-traffic` and the `policy-map` name is `video-monitor`.

**Step 8**

```
class class-map-name
```

Example: router(config-pmap)# class video-class

**Purpose**

Specifies the traffic (class map) on which an action is to be performed.

In this example, the class map is `video-class`.

**Step 9**

```
monitor parameters
```

Example: router(config-pmap-c)# monitor parameters

**Purpose**

Enters the monitor parameters submode where you can configure the flow related parameters.

**Step 10**

```
df rfc4445 | ipdv
```

Example: router(config-pmap-c-monitor)# df ipdv

**Purpose**

Specifies the jitter buffer calculation mechanism.

**Note** By default, rfc4445 algorithm is selected.

**Step 11**

```
interval duration n-secs
```

Example: router(config-pmap-c-monitor)# interval duration 30

**Purpose**

Specifies the monitoring interval. The loss or jitter of packets is calculated at the end of this interval. The configurable range is 30 to 900 seconds. The default value is 30 seconds. The interval value should be a multiple of 5.

**Step 12**

```
timeout n-inteval
```

Example: router(config-pmap-c-monitor)# timeout 200

**Purpose**

Specifies the timeout for a flow. If no traffic is transmitted within this interval, the monitoring is stopped. When the flow times out, the resources linked with that flow are released. The default value is 100 intervals.

**Step 13**

```
history n-inteval
```

Example: router(config-pmap-c-monitor)# history 20

**Purpose**

Specifies the last `n-interval` number of intervals that should be maintained in the history table. The range is 1 to 180 intervals. The default value is 10 intervals.

**Step 14**

```
exit
```

Example: router(config-pmap-c-monitor)# exit

**Purpose**

Exits the monitor parameter mode.

**Step 15**

```
monitor metric {mdi|ip-cbr|rtp}
```

Example: router(config-pmap-c)# monitor metric mdi

**Purpose**

Enters the monitor metric submode where you can configure the metric related parameters.

In this example, the MDI metric is selected.
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 16</strong></td>
<td><strong>clock-rate</strong> <em>dynamic_pt frequency</em>&lt;br&gt;Example: &lt;br&gt;router(config-pmap-c-metric)# clock-rate 1 96</td>
</tr>
<tr>
<td><strong>Step 17</strong></td>
<td><em>(optional)</em> <strong>monitor pids</strong> <em>pid1 [pid2] [pid3] [pid4] [pid5]</em>&lt;br&gt;Example: &lt;br&gt;router(config-pmap-c-metric)# monitor pids 0x0011</td>
</tr>
<tr>
<td><strong>Step 18</strong></td>
<td><strong>rate media</strong> <em>n</em> *(bps</td>
</tr>
<tr>
<td><strong>Step 19</strong></td>
<td><strong>packet</strong> *(size media n-bytes</td>
</tr>
<tr>
<td><strong>Step 20</strong></td>
<td><strong>exit</strong>&lt;br&gt;Example: &lt;br&gt;router(config-pmap-c-metric)# exit</td>
</tr>
<tr>
<td><strong>Step 21</strong></td>
<td><strong>monitor metric</strong> *(mdi</td>
</tr>
</tbody>
</table>

---

**Note**<br>For metric monitoring, you should configure mdi-metric as *rate media* or ip-cbr metric as *rate layer3*.

---

**Example:**

- Maps a dynamic PT value to the corresponding frequency for each class-map. The available frequency options are:
  - 148.5/1.001Mhz
  - 148.5Mhz
  - 27Mhz
  - Disable

- Specifies the expected media transfer rate. For the media transfer rate, you have to specify the transfer rate unit. The following units are available:
  - **bps**: Number of bits per second
  - **kbps**: Number of kilobits per second
  - **mbps**: Number of megabits per second
  - **gbps**: Number of gigabits per second

- Specifies the layer 2 or layer 3 packet behavior. In general, the keyword *media* refers to layer 2 video or audio frame whereas layer 3 refers to network layer packet such as IP layer packet.

  The keyword *size media* specifies the encoding video or audio frame size in bytes. The valid value is 188.

  The keyword *count media in-layer3* specifies the number of MPEG frames within a single IP packet. The default value is 7 and valid range is 1 - 7.

- Exits the monitor metric mode.

- Enters the monitor metric submode for IP-CBR where you can configure the metric related parameters.
### Inline Video Monitoring on the Cisco 7600 Router

#### Step 22
**rate layer3 packet n [pps]**

Example:
```plaintext
router(config-pmap-c-metric)# rate layer3 packet 300
```

**Purpose:** Specifies the expected layer 3 transfer rate. The transfer rate is configured in packets per second (pps).

For accurate metric computations, recommended pps configuration should be three precision digits.

**Note**  
For metric monitoring, you should configure mdi-metric as `rate media` or ip-cbr metric as `rate layer3`. If both the options are configured, ip-cbr metric configuration takes precedence.

#### Step 23
**exit**

Example:
```plaintext
router(config-pmap-c-metric)# exit
```

**Purpose:** Exits the monitor metric mode.
Step 24  react profile-id {mdi-df | mdi-mdc | mdi-mlr | ip-cbr:mrv | ip-cbr:df | media-stop | rtp | transport-availability | error-seconds}

Configures the react metrics. At the end of the interval, values are compared with the configured threshold values. If the system exceeds these configured values, an alarm is triggered.

This command enters the react submode where you can configure the alarms and threshold values.

When the monitored interval for a flow expires, the corresponding metric values are generated. These values are compared to the threshold values you set here, and if the threshold is crossed, an alarm is exported to the management system.

You can specify multiple react commands. Each command is differentiated by the argument `operation-id value`. The react argument `operation-id value` should be unique within a policy-map. The range of the argument `operational id-value` is 1 to 65535. The react types are:

- mdi-df
- mdi-mdc
- mdi-mlr
- ip-cbr-mrv
- ip-cbr:df
- media-stop
- rtp
- transport-availability
- error-seconds

**Note**  
If you selected the media-stop option, you cannot configure more than one react profile for a class-map for react type `media-stop`.

A `profile-id` once used for a react type cannot be reused for any other react type until it is removed using the `no react profile-id react-type` command.
Step 25

```
threshold {range range-value1 range-value2} | 
{{gt|ge|lt|le} value3 | {type {immediate | average value4}}}
```

Specifies the threshold related parameters.

- **range**: Specifies the threshold range. The unit for this boundary depends on react type in the `react` command. If the react type is MDI:DF, the unit is `msec`. If the react type is MDI:MLR, the unit is `number of packets lost`.

- **gt|ge|lt|le**: Specifies the threshold range where `gt` stands for greater than, `ge` stands for greater than or equal to, `lt` stands for less than, and `le` stands for less than or equal to.

You need to specify one value and threshold parameter.

In the following example, the threshold is between 100 and infinity:

```
threshold range gt 100
```

- **type**: Specifies the criteria for alarm assertion. If you select the keyword `immediate`, an alert is triggered at the end of monitoring interval if the metric value crosses the configured range. If you select the keyword `average`, the alarm is generated based on the average value which is computed based on the `value4` you set. The range of the `value4` is between 2 and the number defined in the flow history.

The default type is `immediate`.

Step 26

```
action {syslog | snmp}
```

Enables the management system to log the threshold-crossing events.

Example:
```
router(config-pmap-c-react)# threshold gt 4
router(config-pmap-c-react)# threshold type average 5
```

Step 27

```
alarm severity {none | informational | notification | warning | error | critical | alert | emergency}
```

Specifies the alarm severity associated with a particular `react` command.

The default value is `none`. The router does not generate syslog message if alarm severity is set to `none`.

Example:
```
router(config-pmap-c-react)# alarm severity none
```

Step 28

```
alarm type discrete
```

Specifies that discrete alarms are supported.

Example:
```
router(config-pmap-c-react)# alarm type discrete
```

**Note**

Alarm groups are not supported for Video Monitoring feature release 15.0(01)S.

Step 29

```
description character-string
```

Adds the comments for the submodes. Available for all the submodes. The `character-string` cannot exceed 200 characters.

Example:
```
router(config-pmap-c-react)# description critical TCA
```

Step 30

```
end
```

Exits the configuration mode.

Example:
```
router(config-pmap-c-react)# end
```

Step 31

```
configure terminal
```

Enters the configuration mode.

Example:
```
router# configure terminal
```
**Chapter 13      Configuring Layer 3 and Layer 4 Features**

**Inline Video Monitoring on the Cisco 7600 Router**

**Example**

The following example shows how to configure video monitoring feature on an interface:

```
Router(config)# policy-map type performance-traffic video-monitor
Router(config-pmap)# class video-class
Router(config-pmap-c)# monitor parameters
Router(config-pmap-c-monitor)# df ipdv
Router(config-pmap-c-monitor)# description mon
Router(config-pmap-c-monitor)# interval duration 30
Router(config-pmap-c-monitor)# history 30
Router(config-pmap-c-monitor)# timeout 10
Router(config-pmap-c-monitor)# exit
Router(config-pmap-c)# monitor metric ip-cbr
Router(config-pmap-c-metric)# rate layer3 packet 237.465 pps
Router(config-pmap-c-metric)# exit
Router(config-pmap-c)# monitor metric mdi
Router(config-pmap-c-metric)# monitor pids 0x0011
Router(config-pmap-c-metric)# rate media 2500031 bps
Router(config-pmap-c-metric)# packet count media in-layer3 7
Router(config-pmap-c-metric)# packet size media 188
Router(config-pmap-c-metric)# exit
Router(config-pmap-c)# react 1 ip-cbr-df
Router(config-pmap-c-react)# alarm severity critical
Router(config-pmap-c-react)# threshold type immediate
Router(config-pmap-c-react)# threshold ge 30.000
Router(config-pmap-c-react)# react 2 ip-cbr-mrv
Router(config-pmap-c-react)# alarm severity informational
Router(config-pmap-c-react)# threshold type immediate
Router(config-pmap-c-react)# threshold range 20.000 50.000
Router(config-pmap-c-react)# react 3 mdi-df
Router(config-pmap-c-react)# alarm severity critical
Router(config-pmap-c-react)# threshold type immediate
Router(config-pmap-c-react)# threshold range 20.000 50.000
Router(config-pmap-c-react)# react 4 mdi-mlr
Router(config-pmap-c-react)# alarm severity critical
Router(config-pmap-c-react)# threshold type immediate
Router(config-pmap-c-react)# threshold gt 0
```

**Command or Action**

<table>
<thead>
<tr>
<th>Step 32</th>
<th><code>interface type number</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>router(config)# interface gig 1/2</code></td>
</tr>
</tbody>
</table>

**Purpose**

Configures the interface type and number.

<table>
<thead>
<tr>
<th>Step 33</th>
<th><em>(optional)</em> <code>service instance instance-number Ethernet</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>router(config)# service instance 1 Ethernet</code></td>
</tr>
</tbody>
</table>

**Purpose**

Configures the service instance for EVC.

**Note**

Applicable while configuring EVC.

| Step 34 | *(optional)* `service-policy type performance-traffic (input|output) policy-map-name` |
|---------|---------------------------------------------------------------------|
| Example:  | `router(config-if-srv)# service-policy type performance-traffic input video-monitor` |

**Purpose**

Attaches the specified `policy-map` to the target EVC.

**Note**

Applicable while configuring EVC.

<table>
<thead>
<tr>
<th>Step 35</th>
<th><code>exit</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>router(config-if)# exit</code></td>
</tr>
</tbody>
</table>

**Purpose**

Exits the interface configuration mode.
Inline Video Monitoring on the Cisco 7600 Router

Chapter 13  Configuring Layer 3 and Layer 4 Features

Router(config-pmap-c-react)#react 5 media-stop
Router(config-pmap-c-react)#description for me
Router(config-pmap-c-react)#alarm severity critical
Router(config-pmap-c-react)#react 15 mdi-mdc
Router(config-pmap-c-react)#alarm severity notification
Router(config-pmap-c-react)#threshold type immediate
Router(config-pmap-c-react)#threshold gt 0
Router(config-pmap-c-react)#react 10 ip-cbr-mrv
Router(config-pmap-c-react)#alarm severity informational
Router(config-pmap-c-react)#threshold type immediate
Router(config-pmap-c-react)#threshold ge 1.00000
Router(config-pmap-c-react)#exit
Router(config-pmap-c)#exit
Router(config-pmap)#exit
Router(config)#interface TenGigabitEthernet3/1
Router(config-if)#service-policy type performance-traffic input video-monitor
Router(config-if)#end

This example shows how to configure RTP metrics for video monitoring:

Router(config)#policy-map type performance-traffic video-monitor
Router(config-pmap)#class video-class
Router(config-pmap-c)# monitor parameters
Router(config-pmap-c-monitor)#description mon
Router(config-pmap-c-monitor)#interval duration 30
Router(config-pmap-c-monitor)#history 30
Router(config-pmap-c-monitor)#timeout 10
Router(config-pmap-c-monitor)#exit
Router(config-pmap-c)#monitor metric rtp
Router(config-pmap-c-metric)#exit
Router(config-pmap-c)#react 1 rtp-jitter
Router(config-pmap-c-react)#alarm severity critical
Router(config-pmap-c-react)#threshold type immediate
Router(config-pmap-c-react)#threshold ge 30.000
Router(config-pmap-c-react)#react 2 rtp-loss-rate
Router(config-pmap-c-react)#alarm severity informational
Router(config-pmap-c-react)#threshold type immediate
Router(config-pmap-c-react)#threshold le 50.00
Router(config-pmap-c-react)#react 3 rtp-max-jitter
Router(config-pmap-c-react)#alarm severity critical
Router(config-pmap-c-react)#threshold type immediate
Router(config-pmap-c-react)#threshold range 20.000 50.000
Router(config-pmap-c-react)#react 4 rtp-lost-pkts
Router(config-pmap-c-react)#alarm severity critical
Router(config-pmap-c-react)#threshold type immediate
Router(config-pmap-c-react)#threshold ge 10
Router(config-pmap-c-react)#react 5 media-stop
Router(config-pmap-c-react)#description for me
Router(config-pmap-c-react)#alarm severity critical
Router(config-pmap-c-react)#exit
Router(config-pmap-c)#exit
Router(config-pmap)#exit
Router(config)#interface TenGigabitEthernet3/1
Router(config-if)#service-policy type performance-traffic input video-monitor
Router(config-if)#end

Verifying the Configuration

Use the show policy-map type performance-traffic interface interface-name command to display all the flows learnt on the specified interface.

- Output for IPCBR/MDI:
Router#show policy-map type performance-traffic interface gig 8/11
GigabitEthernet8/11

Service-policy input: video-swport

```
class-map: sw-vlan3
------------------------------------------------------------------------------------------
Mon-Interval(sec): 30, History(intvls): 5, Timeout(sec): 60, DF: rfc4445, Total Flows: 1
------------------------------------------------------------------------------------------
Flow: 0001, IPV4; Dest: 12.0.0.2 Port: 6300; Src: 11.0.0.2 Port: 63
Agg Value(Per Flow)

<table>
<thead>
<tr>
<th>MDC</th>
<th>Avail(%)</th>
<th>Pkt_cnt</th>
<th>DF(msec)</th>
<th>MDC</th>
<th>Seconds</th>
<th>Avail (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25200</td>
<td>100.000</td>
<td>126002</td>
<td>10.075</td>
<td>NA</td>
<td>NA</td>
<td>100.000</td>
</tr>
</tbody>
</table>
```

```
Error Transport

<table>
<thead>
<tr>
<th>Intvl Updated at Type</th>
<th>Pkt_cnt</th>
<th>MRV(%)/MLR</th>
<th>DF(msec)</th>
<th>MDC</th>
<th>Seconds</th>
<th>Avail (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cbr</td>
<td>3000</td>
<td>0.00000</td>
<td>10.075</td>
<td>NA</td>
<td>NA</td>
<td>100.000</td>
</tr>
<tr>
<td>mdi</td>
<td>3000</td>
<td>0.00000</td>
<td>10.075</td>
<td>NA</td>
<td>NA</td>
<td>100.000</td>
</tr>
<tr>
<td>cbr</td>
<td>3000</td>
<td>0.00000</td>
<td>10.075</td>
<td>NA</td>
<td>NA</td>
<td>100.000</td>
</tr>
<tr>
<td>mdi</td>
<td>3000</td>
<td>0.00000</td>
<td>10.075</td>
<td>NA</td>
<td>NA</td>
<td>100.000</td>
</tr>
<tr>
<td>cbr</td>
<td>3001</td>
<td>0.00000</td>
<td>10.075</td>
<td>NA</td>
<td>NA</td>
<td>100.000</td>
</tr>
<tr>
<td>mdi</td>
<td>3001</td>
<td>0.00000</td>
<td>10.075</td>
<td>NA</td>
<td>NA</td>
<td>100.000</td>
</tr>
</tbody>
</table>

```

Output for RTP:

Router#show policy-map type performance-traffic interface gig 8/11 in class sw$
GigabitEthernet8/11

Service-policy input: video-swport

```
class-map: sw-rtp-vlan3
------------------------------------------------------------------------------------------
Mon-Interval(sec): 30, History(intvls): 5, Timeout(sec): 60, Total Flows: 1
------------------------------------------------------------------------------------------
Flow: 0001, IPV4; Dest: 12.0.0.2 Port: 6300; Src: 11.0.0.2 Port: 63
Agg Value(Per Flow)

<table>
<thead>
<tr>
<th>MDC</th>
<th>Avail(%)</th>
<th>Pkt_cnt</th>
<th>DF(msec)</th>
<th>MDC</th>
<th>Seconds</th>
<th>Avail (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25200</td>
<td>100.000</td>
<td>126002</td>
<td>10.075</td>
<td>NA</td>
<td>NA</td>
<td>100.000</td>
</tr>
</tbody>
</table>
```

```
Error Transport

<table>
<thead>
<tr>
<th>Intvl Updated at Type</th>
<th>Pkt_cnt</th>
<th>MRV(%)/MLR</th>
<th>DF(msec)</th>
<th>MDC</th>
<th>Seconds</th>
<th>Avail (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtp</td>
<td>3000</td>
<td>0.00000</td>
<td>10.075</td>
<td>NA</td>
<td>NA</td>
<td>100.000</td>
</tr>
<tr>
<td>rtp</td>
<td>30536</td>
<td>1.75530</td>
<td>0.007</td>
<td>0.000 100.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
Note
```
Video-monitoring on ethernet service instance is supported on ScEompls, SVI based Eompls, VPLS, EVC BD, and EVC local connect services.
Use the `show policy-map type performance-traffic interface interface_name aggregate` command to display the total number of flows on an interface:

Router#show policy-map type performance-traffic interface gig 8/11 aggregate
GigabitEthernet8/11

Service-policy input: video-swport
Total Number of flows : 6

Use the `show policy-map type performance-traffic interface interface_name brief` command to display brief description of all the metrics for all the flows on an interface.

Router#show policy-map type performance-traffic interface gig 8/11 brief
GigabitEthernet8/11

Service-policy input: video-swport

class-map: sw-vlan1

---

Mon-Interval(sec): 30, History(intvls): 2, Timeout(sec): 60, DF: rfc4445, Total Flows: 1
---

<table>
<thead>
<tr>
<th>FlowID</th>
<th>Flow Key</th>
<th>MRV(%)</th>
<th>MDC</th>
<th>MLR</th>
<th>Error</th>
<th>Transport Avail(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21.0.1.2:63-&gt;32.0.1.2:5000,10:0</td>
<td>0.00000</td>
<td>32400</td>
<td>32400</td>
<td>0.000</td>
<td>100.000</td>
</tr>
</tbody>
</table>

class-map: sw-rtp-vlan1

---

Mon-Interval(sec): 30, History(intvls): 3, Timeout(sec): 60, Total Flows: 1
---

<table>
<thead>
<tr>
<th>FlowID</th>
<th>Flow Key</th>
<th>Exp</th>
<th>Lost</th>
<th>Pkts</th>
<th>Rate(%)</th>
<th>Jitter</th>
<th>Lost</th>
<th>Intvls</th>
<th>Err</th>
<th>Transport Avail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21.0.1.3:63-&gt;32.0.1.2:50000,10:0</td>
<td>1633428</td>
<td>13400</td>
<td>25</td>
<td>0.200</td>
<td>99.975</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Use the `show policy-map type performance-traffic interface interface_name cumulative` command to display cumulative metrics for the flows on a specified interface.

Router#show policy-map type performance-traffic interface gig 8/11 cumulative
GigabitEthernet8/11

Service-policy input: video-swport

class-map: sw-vlan1

---

Mon-Interval(sec): 30, History(intvls): 2, Timeout(sec): 60, DF: rfc4445, Total Flows: 1
---

<table>
<thead>
<tr>
<th>FlowID</th>
<th>Flow Key</th>
<th>MDC</th>
<th>NLR</th>
<th>Error</th>
<th>Secs</th>
<th>Avail(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21.0.1.2:63-&gt;32.0.1.2:50000,10:0</td>
<td>0.00000</td>
<td>32400</td>
<td>32400</td>
<td>0.000</td>
<td>100.000</td>
</tr>
</tbody>
</table>

class-map: sw-rtp-vlan1

---

Mon-Interval(sec): 30, History(intvls): 3, Timeout(sec): 60, Total Flows: 1
---

<table>
<thead>
<tr>
<th>FlowID</th>
<th>FlowKey</th>
<th>Exp</th>
<th>Lost</th>
<th>Pkts</th>
<th>Intvls</th>
<th>Resyncs</th>
<th>Err</th>
<th>Avail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21.0.1.3:63-&gt;32.0.1.2:50000,10:0</td>
<td>1633428</td>
<td>13400</td>
<td>25</td>
<td>25</td>
<td>0.200</td>
<td>99.975</td>
<td></td>
</tr>
</tbody>
</table>
Use the `show policy-map type performance-traffic interface interface_name input|output` command to display the data flow on an interface in a specified direction.

Router#show policy-map type performance-traffic interface gig 8/11 input

Service-policy input: video-swport

class-map: sw-vlan3

Mon-Interval(sec): 30, History(intvls): 5, Timeout(sec): 60, DF: rfc4445, Total Flows: 1

Flow: 0001, IPV4; Dest: 12.0.0.2 Port: 63000; Src: 11.0.0.2 Port: 63

<table>
<thead>
<tr>
<th>MDC</th>
<th>Available</th>
<th>Pkt_cnt</th>
<th>Error SECs</th>
<th>MRV</th>
<th>MDC Seconds</th>
<th>Avail (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>37200</td>
<td>100.000</td>
<td>186003</td>
<td>0.00000</td>
<td>0.000000</td>
<td>NA</td>
<td>100.000</td>
</tr>
</tbody>
</table>

Use the `show policy-map type performance-traffic interface interface_name detail` command to display the detailed information for the latest interval of each flow.

Router#show policy-map type performance-traffic interface gig 8/11 detail

Service-policy input: video-swport
class-map: sw-vlan3

------------------------------------------------------------------------------------------
Mon-Interval(sec): 30, History(intvls): 5, Timeout(sec): 60, DF: rfc4445, Total Flows: 1
------------------------------------------------------------------------------------------

Flow: 0001 Key: 11.0.0.2:63 -> 12.0.0.2:6300  Intervals : 1
Intvl# 68, Updated at 21:34:06.775 PDT Fri Jun 10 2011
Metric Type  : IP-CBR
MRV          : 0.00000%  DF(ms) : 10.075
Packets      : 3000  Bytes  : 4296000

Intvl# 68, Updated at 21:34:06.775 PDT Fri Jun 10 2011
Metric Type  : MDI
MLR          : 600  MDC    : 600
Packets      : 3000  Bytes  : 4296000
DF(ms)       : 10.075  Error seconds : 0.000
Transport Availability (%) : 100.000

class-map: sw-rtp-vlan3

------------------------------------------------------------------------------------------
---
Mon-Interval(sec): 30, History(intvls): 5, Timeout(sec): 60, Total Flows: 1
---

Flow: 0001 Key: 11.0.0.12:5000 -> 12.0.0.2:50000, 3735374711  Intervals : 1
Intvl# 68, Updated at 21:34:01.731 PDT Fri Jun 10 2011
Pkts Recieved : 30000         Pkts Exp    : 30536         Pkts Valid       : 30000
Pkts Lost     : 536           Pkts Late   : 0             Pkts reord       : 0
Loss Rate (%) : 1.75530       Loss Intvls : 1             Avg Loss duration: 536.00
Jitter(msec)  : 0.006         Max Jitter : 0.024         Resyncs          : 1
Error seconds : 0.008         Transport Availability (%) : 99.973

Use the show policy-map type performance-traffic interface interface_name last n command to
display the last n number of intervals for each flow on an interface:

Router#show policy-map type performance-traffic interface gig 8/11 last 2
GigabitEthernet8/11

Service-policy input: video-swport

class-map: sw-vlan3

------------------------------------------------------------------------------------------
Mon-Interval(sec): 30, History(intvls): 5, Timeout(sec): 60, DF: rfc4445, Total Flows: 1
------------------------------------------------------------------------------------------

Flow: 0001, IPV4;   Dest: 12.0.0.2  Port: 6300;   Src: 11.0.0.2  Port: 63
Agg Value(Per Flow)
MDC : 39600          Avail(%) : 100.000     Pkt_cnt : 198003
MLR : 39600          Error_secs : 0.000    MRV(%) : 0.00000

<table>
<thead>
<tr>
<th>Intvl</th>
<th>Updated at</th>
<th>Type</th>
<th>Pkt_cnt</th>
<th>MRV(%)</th>
<th>MLR(%)</th>
<th>MDC</th>
<th>Error Seconds</th>
<th>Transport Availability (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
<td>21:34:36</td>
<td>cbr</td>
<td>3000</td>
<td>0.0000</td>
<td>10.075</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>69</td>
<td>21:34:36</td>
<td>mdi</td>
<td>3000</td>
<td>0.0000</td>
<td>10.075</td>
<td>600</td>
<td>0.000</td>
<td>100.000</td>
</tr>
<tr>
<td>68</td>
<td>21:34:06</td>
<td>cbr</td>
<td>3000</td>
<td>0.0000</td>
<td>10.075</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>68</td>
<td>21:34:06</td>
<td>mdi</td>
<td>3000</td>
<td>0.0000</td>
<td>10.075</td>
<td>600</td>
<td>0.000</td>
<td>100.000</td>
</tr>
</tbody>
</table>

class-map: sw-rtp-vlan3
Use the `show policy-map type performance-traffic interface interface-name service instance instance-number` command to display all the flows learnt on the specified EVC:

Router#show policy-map type performance-traffic interface gig 8/11 ser in 1

GigabitEthernet8/11: EFP 1
Service-policy input: video-monitor

class-map: mpls

Router#show policy-map type performance-traffic interface gig 8/11 ser in 1

GigabitEthernet8/11: EFP 1
Service-policy input: video-monitor

class-map: mpls

Router#show policy-map type performance-traffic interface gig 8/11 ser in 1

GigabitEthernet8/11: EFP 1
Service-policy input: video-monitor

class-map: mpls
Use `show running-config interface interface-name` command to display detailed information about interface:

```
router#sh running-config interface tenGigabitEthernet 7/21
Building configuration...
Current configuration : 816 bytes
interface TenGigabitEthernet7/21
  ip arp inspection limit none
  no ip address
  ip rsvp bandwidth
  service instance 1 ethernet
  encapsulation dot1q 101
  rewrite ingress tag pop 1 symmetric
  service-policy type performance-traffic input video_monitor_1
  service-policy type performance-traffic output video_monitor_2
  bridge-domain 101
  service instance 2 ethernet
  encapsulation dot1q 102
  rewrite ingress tag pop 1 symmetric
  service-policy type performance-traffic input video_monitor_1
  service-policy type performance-traffic output video_monitor_2
  bridge-domain 102
end
```

Use the `show policy-map type performance-traffic interface interface_name match ipv4 source ip-address mask destination ip-address mask` command to display the flow matching the specified IPV4 source or destination IP:

```
Router#show policy-map type performance-traffic interface gig 8/11 match ipv4 source 11.0.0.12 255.255.255.255 destination 12.0.0.2 255.255.255.255
GigabitEthernet8/11
Service-policy input: video-swport
  class-map: sw-vlan3
--------------------------------------------------------------------------
  Mon-Interval(sec): 30, History(intvls): 5, Timeout(sec): 60, DF: rfc4445, Total Flows: 1
--------------------------------------------------------------------------
  class-map: sw-rtp-vlan3
--------------------------------------------------------------------------
  Mon-Interval(sec): 30, History(intvls): 5, Timeout(sec): 60, Total Flows: 1
--------------------------------------------------------------------------
Flow: 0001, IPV4;  Dest: 12.0.0.2  Port: 50000;  Src: 11.0.0.12  Port: 5000; rtp-ssrc: 3735927471
  Agg Value(Per Flow)
  Avail(%) : 99.973  Loss_Intvls : 32  Resyncs : 32
  Pkt_cnt   : 2130038  Pkt_exp     : 2147190  Pkt_lost  : 17152
  Error_secs : 0.256  Pkt   Exp  Lost  Loss  Jitter  MaxJitter Avg. Loss  Loss  Err Transport
```
### Troubleshooting the Inline Video Monitoring Implementation

The following section describes how to troubleshoot Video Monitoring.

- **Flow is not displayed in the show command**
  
  A flow is defined as unique traffic identified by the source and destination IP and port information. When the flow path is not displayed by the `show` command, perform the following steps to identify the problem:

1. Check the interface statistics using the `show interface interface-type slot/port` command to ensure that the traffic is flowing.

2. Check the configuration of class-map and the ACL configured under the class-map to ensure that the ACL is classifying the flows. The following example shows how to check the configuration of a class-map:

   ```
   ROUTER#show running-config class-map video-class
   Building configuration...
   Current configuration : 67 bytes
   !
   class-map match-any video-class
     match access-group 102
   !
   end
   ROUTER#
   ```

3. Check whether the `rate layer3 packet` command or `rate media` command is configured under the class using `show policy-map type performance-traffic policy-map-name` command.

   **Note** The data flow path is not learnt for fragmented packets, MPLS packets, non-UDP protocols, and tunneled packets.

- **The change in media rate does not affect the DF metrics.**
  
  Use the `show policy-map type performance-traffic policy-map-name` command to check if the `rate layer3 packet` command is configured for the class. If the `rate layer3 packet` command is configured for the class, the IP-CBR packet rate configuration is used for both the IP-CBR and MDI metric calculations.
DF value is returned even though the data flow stream is stable. DF is used to determine the jitter buffer required to ensure effective utilization of buffer while handling a stream. The minimum jitter buffer size is sufficient to receive a single packet. Therefore, even when there is no impairment or delay in the stream, the DF is equal to an inter-packet-gap. This DF value reported by the router when there is no impairment, is approximately equal to 1/packet-rate.

When packets are dropped, no message is triggered for MDI:DF even if the TCAs (reacts) are configured for MDI:DF. When there are drops seen in the stream, DF computed is incorrect. In such a case, where packets are dropped in a stream (MLR), the computed DF is not used for triggering the message.

Show command output returns a dash (-).

Indicates that the metrics computed for that interval are invalid. This condition occurs during the initial flow learn, when a policy-map is updated dynamically or when the next intervals on the MSE are reported for the current interval.

Metrics cannot be configured under the default class (class-default).

Performance traffic functionality is not supported in the default class. The default class includes the traffic that is not classified under any other class-maps and has no defined rate. It is not possible to configure metric parameters for the default class.

TCA threshold messages are not triggered even when the metric value crosses the configured range. Use the show policy-map type performance-traffic policy-map-name command to verify that the alarm severity is not configured to none.

Uncertainty over the choice of right debug logs.

Complete the following steps to collect the output for the line card:

1. Run the attach module-number command to connect to the line card.
2. Run the show platform npc performance-traffic action np number interface classmap command to display the class-map configuration on the line card.
3. Run the show platform npc performance-traffic action np number interface result command to display the class-map structure used by the microcode.
4. Run the show platform npc performance-traffic action np number interface stats command to print per flow statistics for the network processor.
5. Run the show platform npc performance-traffic action np number stats command to print the aggregate flow count in the network processor.
6. Run the show platform npc performance-traffic classification all to print the classification details for each class.

**Supported MIBs**

Video Monitoring supports the following MIBs. These MIBs are used for retrieving the data collected by flow monitors.

- **CISCO-FLOW-MONITOR-TC-MIB**: This MIB module defines the text conventions common to the rest of the MIB modules.
- **CISCO-FLOW-MONITOR-MIB**: This MIB module defines a framework that describes the flow monitors supported by the system, the flows that are learned, and the flow metrics collected for those flows.
IP Tunneling - IPv6 Rapid Deployment

The following sections describe the IPv6 Rapid Deployment (6RD) function.

- Understanding IPv6 Rapid Deployment, page 13-31
- Restriction for IPv6 Rapid Deployment., page 13-33
- Configuring IPv6 Rapid Deployment on the Cisco 7600 series router Platform, page 13-34
- Troubleshooting Tips, page 13-37

Understanding IPv6 Rapid Deployment

The 6RD deployment is an variant of the 6to4 feature, and allows a service provider to provide a unicast IPv6 service to customers over its IPv4 network (using IPv6 encapsulation in IPv4).


The differences between 6RD and 6to4 tunneling are:

- 6RD does not require IP addresses to have a 2002::/16 prefix. Therefore, the prefix can be from the service provider’s own address block. This function allows the 6RD operational domain to be within the service provider network. From the perspective of customer sites and the general IPv6 internet connected to a 6RD-enabled service provider network, the IPv6 service provided is equivalent to native IPv6.

- Not all the 32 bits from the IPv4 destination address are carried to the IPv6 payload header. The IPv4 destination is obtained from a combination of bits in the payload header and information on the router. The IPv4 address is not at a fixed location in the IPv6 header as in the case with 6to4 tunneling.

Figure 13-1 shows a high-level view of the 6RD deployment.
IP Tunneling - IPv6 Rapid Deployment

Chapter 13 Configuring Layer 3 and Layer 4 Features

Figure 13-1 6RD Deployment

The service provider delegates a 6RD service provider prefix for the IPv6 deployment, using the IPv4 address bits.

Figure 13-2 shows how 6RD prefix delegation works.

Figure 13-2 6RD Prefix Delegation

IPv4 common prefix 10.1.0.0/16
IPv4 common suffix 0.0.0.1/8

0 to 32 bits 0 to 32 bits 64 bits

SP Prefix IPv4 Address Bits Subnet ID Interface ID

6RD delegated prefix

Users’ Address Space

IPv4 Destination Address

16 bits 8 bits 8 bits

IPv4 Common Prefix IPv4 Address Bits IPv4 common Suffix
Figure 13-3 shows the 6RD prefix delegation topology.

**Figure 13-3  6RD Prefix Delegation Topology**

![6RD Prefix Delegation Topology Diagram]

<table>
<thead>
<tr>
<th>SP Prefix</th>
<th>2001:B000::/32</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 Common Prefix</td>
<td>10.1.0.0/16</td>
</tr>
<tr>
<td>IPv4 Common Suffix</td>
<td>0.0.0.1/8</td>
</tr>
<tr>
<td>CE1: Delegated 6RD prefix</td>
<td>2001:B000:0100::/40</td>
</tr>
<tr>
<td>CE2: Delegated 6RD prefix</td>
<td>2001:B000:0200::/40</td>
</tr>
<tr>
<td>BR: Delegated 6RD prefix</td>
<td>2001:B000:0400::/40</td>
</tr>
<tr>
<td>CE1 (IPv4) tunnel transport source</td>
<td>10.1.1.1</td>
</tr>
<tr>
<td>CE2 (IPv4) tunnel transport source</td>
<td>10.1.2.1</td>
</tr>
<tr>
<td>BR (IPv4) tunnel transport source</td>
<td>10.1.4.1</td>
</tr>
</tbody>
</table>

**Restriction for IPv6 Rapid Deployment.**

The interface facing the IPv4 network must be on the ES40 linecard.

**Supported Features**

Table 13-3 shows the list of supported and unsupported features for 6RD functionality.

<table>
<thead>
<tr>
<th>Table 13-3 Supported and Unsupported Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td>6RD BR mode</td>
</tr>
<tr>
<td>6RD CE mode</td>
</tr>
<tr>
<td>6RD tunnel</td>
</tr>
<tr>
<td>Scale</td>
</tr>
</tbody>
</table>
Configuring IPv6 Rapid Deployment on the Cisco 7600 series router Platform

The following sections describe how to configure 6RD on the c7600 platform:

- Configuring 6RD, page 13-34
- Verifying the Configuration, page 13-37

Configuring 6RD

Complete the following steps to configure 6RD.

SUMMARY STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>enable</td>
</tr>
<tr>
<td>2</td>
<td>configure terminal</td>
</tr>
<tr>
<td>3</td>
<td>interface tunnel tunnel-number</td>
</tr>
<tr>
<td>4</td>
<td>ipv6 address {ipv6-address/prefix-length</td>
</tr>
<tr>
<td>5</td>
<td>tunnel source {ip-address</td>
</tr>
<tr>
<td>6</td>
<td>tunnel mode ipv6ip [6rd</td>
</tr>
<tr>
<td>7</td>
<td>mls 6rd reserve interface gigabitethernet/ tengigabitethernet</td>
</tr>
<tr>
<td>8</td>
<td>tunnel 6rd prefix ipv6-prefix/prefix-length</td>
</tr>
<tr>
<td>9</td>
<td>tunnel 6rd ipv4 {prefix-length length} {suffix-length length}</td>
</tr>
<tr>
<td>10</td>
<td>exit</td>
</tr>
<tr>
<td>11</td>
<td>interface type instance</td>
</tr>
<tr>
<td>12</td>
<td>ip address ip-address</td>
</tr>
<tr>
<td>13</td>
<td>exit</td>
</tr>
<tr>
<td>14</td>
<td>ipv6 route {ipv6-prefix</td>
</tr>
<tr>
<td>15</td>
<td>end</td>
</tr>
</tbody>
</table>

### Feature Table

<table>
<thead>
<tr>
<th>Feature</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIBs</td>
<td>No</td>
</tr>
<tr>
<td>Linecards</td>
<td>ES40</td>
</tr>
<tr>
<td>VRF awareness</td>
<td>Yes</td>
</tr>
<tr>
<td>ISG Co-existence</td>
<td>No</td>
</tr>
<tr>
<td>Qos on Tunnels</td>
<td>No</td>
</tr>
</tbody>
</table>

MIBs No
Linecards ES40
VRF awareness Yes
ISG Co-existence No
Qos on Tunnels No
# IP Tunneling - IPv6 Rapid Deployment

## Chapter 13      Configuring Layer 3 and Layer 4 Features

### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**  
   *enable*  
   Example:  
   Router# enable | Enables privileged EXEC mode. Enter your password if prompted. |
| **Step 2**  
   *configure terminal*  
   Example:  
   Router# configure terminal | Enters global configuration mode. |
| **Step 3**  
   *interface tunnel*  
   *tunnel-number*  
   Example:  
   Router(config)# interface tunnel 1 | Specifies a tunnel interface and enters the interface configuration mode. |
| **Step 4**  
   *ipv6 address*  
   *(ipv6-address/prefix-length | prefix-name  
   sub-bits/prefix-length)*  
   Example:  
   Router(config-if)# ipv6 address 2001:B000:400::1/124 | Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface. |
| **Step 5**  
   *tunnel source*  
   *(ip-address | interface-type  
   interface-number)*  
   Example:  
   Router(config-if)# tunnel source loopback 0 | Specifies the source interface type and number for the tunnel interface. |
| **Step 6**  
   *tunnel mode ipv6ip*  
   *(6rd | 6to4 | auto-tunnel | isatap)*  
   Example:  
   Router(config-if)# tunnel mode ipv6ip 6rd | Configures a static IPv6 tunnel interface. |
| **Step 7**  
   *mls 6rd reserve interface*  
   *gigabitethernet/  
   tengigabitethernet*  
   Example:  
   Router(config-if)# mls 6rd reserve interface gig 9/5 | Redirects the IPv6 traffic to IPv4 core facing interface on the ES40 line card. |
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td><strong>tunnel 6rd prefix ipv6-prefix/prefix-length</strong></td>
<td>Specifies the common IPv6 prefix on IPv6 rapid 6RD tunnels.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>`Router(config-if)# tunnel 6rd prefix 2001:B000::/32`</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><strong>tunnel 6rd ipv4 {prefix-length length} {suffix-length length}</strong></td>
<td>Specifies the prefix and suffix length of the IPv4 transport address common to all the 6RD routers in a domain.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>`Router(config-if)# tunnel 6rd ipv4 prefix-len 16 suffix-len 8`</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><strong>exit</strong></td>
<td>Exits configuration mode, and returns the CLI to privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>`Router(config-if)# exit`</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><strong>interface type instance</strong></td>
<td>Enters interface configuration mode and names the new loopback interface.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>`Router(config)# interface loopback 0`</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td><strong>ip address ip-address</strong></td>
<td>Assigns an IP address and subnet mask to the loopback interface.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>`Router(config-if)# ip address 10.1.4.1 255.255.255.255`</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td><strong>exit</strong></td>
<td>Exits configuration mode, and returns the CLI to privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>`Router(config-if)# exit`</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td><strong>ipv6 route ipv6-prefix/prefix-length tunnel tunnel-number</strong></td>
<td>Redirects 6RD specific traffic to the 6RD tunnel.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>`Router(config)# ipv6 route 2001:b000::/32 tunnel 1`</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><strong>end</strong></td>
<td>Ends the current configuration session.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>`Router(config-if)# end`</td>
<td></td>
</tr>
</tbody>
</table>
Configuration Examples

This example shows how to configure 6RD.

```
Router# enable
Router# configure terminal
Router(config)# interface tunnel 1
Router(config-if)# ipv6 address 2001:B000:400::1/124
Router(config-if)# tunnel source loopback 0
Router(config-if)# tunnel mode ipv6ip 6rd
Router(config-if)# mls 6rd reserve interface gig 9/5
Router(config-if)# tunnel 6rd prefix 2001:B000::/32
Router(config-if)# tunnel 6rd ipv4 prefix-len 16 suffix-len 8
Router(config-if)# exit
Router(config)# interface loopback 0
Router(config-if)# ip address 10.1.4.1 255.255.255.255
Router(config-if)# exit
Router(config)# ipv6 route 2001:b000::/32 tunnel 1
Router(config)# end
```

Verifying the Configuration

Use these commands to verify the configuration of 6RD on the Cisco 7600 series router:

```
Router# show tunnel 6rd tunnel 10
    Interface Tunnel10:  
        Tunnel Source: 10.1.4.1
        6RD: Operational, V6 Prefix: 2001:B000::/32
        V4 Prefix, Length: 16, Value: 10.1.0.0
        V4 Suffix, Length: 8, Value: 0.0.0.1
        General Prefix: 2001:B000:400::/40

Router# show tunnel 6rd destination 2001:b000:800::12 tunnel 10
    Interface: Tunnel10
    6RD Prefix: 2001:B000:800::12
    Destination: 10.1.8.1

Router# show tunnel 6rd prefix 10.1.8.1 tunnel 10
    Interface: Tunnel10
    Destination: 10.1.8.1
    6RD Prefix: 2001:B000:800::
```

Troubleshooting Tips

For troubleshooting information, contact Cisco Technical Assistance Center (TAC) at:
VRF aware IPv6 Rapid Deployment (6RD) tunnels

Currently the 6RD tunneling feature on c7600 does not support virtual routing and forwarding (VRF) awareness. The forwarding table lookups for IPv6 overlay addresses and IPv4 transport addresses are performed in global routing tables. This feature extends the 6RD tunneling support for IPv6 overlay addresses and IPv4 transport addresses in VRF.

These scenarios explain the VRF aware 6RD tunnel function:

- IPv6 overlay address in VRF and IPv4 transport address in Global routing table (RT).
- IPv6 overlay address in VRF and IPv4 transport address in VRF.

**Figure 13-4** Topology for the IPv6 overlay address in VRF, and the IPv4 transport address in GRT.

The VRF Aware IPv6 over IPv4 Tunnel should have an ES+ line card towards the tunnel facing side.

**Restriction for VRF aware 6RD tunnels**

- Currently the c7600 supports only 256 VRF instances for IPv6.
- The incoming physical interface, and the tunnel interface should have the same VRF instance defined.
- The tunnel transport VRF and the egress physical interface, through which the traffic leaves should have the same VRF instance defined.
Configuring VRF aware 6RD tunnel

The following sections describe how to configure VRF aware IPv6 tunnel on c7600:

- Configuring IPv6 Overlay Addresses in VRF and IPv4 Transport Addresses in Global RT, page 13-39
- Configuring IPv6 overlay addresses in VRF and IPv4 transport addresses in VRF, page 13-47

Configuring IPv6 Overlay Addresses in VRF and IPv4 Transport Addresses in Global RT

Complete the following steps to configure IPv6 overlay addresses in VRF and IPv4 transport addresses in Global RT:

**SUMMARY STEPS**

6RD customer edge router configuration

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>enable</td>
</tr>
<tr>
<td>2</td>
<td>configure terminal</td>
</tr>
<tr>
<td>3</td>
<td>ipv6 unicast-routing</td>
</tr>
<tr>
<td>4</td>
<td>mls ipv6 vrf</td>
</tr>
<tr>
<td>5</td>
<td>vrf definition vrf name</td>
</tr>
<tr>
<td>6</td>
<td>rd {ASN:nn \ IP address: nn}</td>
</tr>
<tr>
<td>7</td>
<td>route-target [import</td>
</tr>
<tr>
<td>8</td>
<td>address-family ipv6</td>
</tr>
<tr>
<td>9</td>
<td>exit</td>
</tr>
<tr>
<td>10</td>
<td>address-family ipv4</td>
</tr>
<tr>
<td>11</td>
<td>exit</td>
</tr>
<tr>
<td>12</td>
<td>exit</td>
</tr>
<tr>
<td>13</td>
<td>interface gigabitethernet slot/port</td>
</tr>
<tr>
<td>14</td>
<td>vrf forwarding vrf name</td>
</tr>
<tr>
<td>15</td>
<td>ipv6 address {ipv6-address/prefix-length \ prefix-name sub-bits/prefix-length}</td>
</tr>
<tr>
<td>16</td>
<td>exit</td>
</tr>
<tr>
<td>17</td>
<td>interface gigabitethernet slot/port</td>
</tr>
<tr>
<td>18</td>
<td>ip address ip-address mask</td>
</tr>
<tr>
<td>19</td>
<td>ip ospf process-id area area-id</td>
</tr>
<tr>
<td>20</td>
<td>exit</td>
</tr>
<tr>
<td>21</td>
<td>interface loopback interface-number</td>
</tr>
</tbody>
</table>
VRF aware IPv6 Rapid Deployment (6RD) tunnels

Step 22  ip address  ip-address  mask
Step 23  ip ospf  process-id  area  area-id
Step 24  exit
Step 25  interface  tunnel  tunnel-number
Step 26  vrf forwarding  vrf  name
Step 27  ipv6 address  {ipv6-address/prefix-length | prefix-name  sub-bits/prefix-length}
Step 28  mls 6rd reserve  interface  gigabitethernet/ tengigabitethernet
Step 29  tunnel source  {ip-address  | interface-type  interface-number}
Step 30  tunnel mode ipv6ip  {6rd  | 6to4  | auto-tunnel  | isatap}
Step 31  tunnel 6rd ipv4  {prefix-length  length}  {suffix-length  length}
Step 32  tunnel 6rd prefix  ipv6-prefix/prefix-length
Step 33  tunnel 6rd br  ipv4-address
Step 34  exit
Step 35  ipv6 route vrf  vrf-name  ipv6-prefix/prefix-length  tunnel  tunnel number
Step 36  ipv6 route vrf  vrf-name  ipv6-prefix/prefix-length  tunnel  tunnel number  ipv6-address
Step 37  end

6RD Border Relay (BR) Router Configuration

Step 1  Repeat steps 1 through 32 from the 6RD CE configuration, and then continue with these steps:
Step 2  exit
Step 3  ipv6 route vrf  vrf-name  ipv6-prefix/prefix-length  tunnel  tunnel number
Step 4  end
### DETAILED STEPS (for 6RD customer edge router configuration)

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td><code>ipv6 unicast-routing</code></td>
<td>Enables the forwarding of IPv6 unicast datagrams.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# ipv6 unicast-routing</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td><code>mls ipv6 vrf</code></td>
<td>Enables IPv6 globally in a VRF instance.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# mls ipv6 vrf</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td><code>vrf definition vrf name</code></td>
<td>Configures a VRF instance and enters the VRF configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# vrf definition VRF_RED</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td></td>
</tr>
<tr>
<td>`rd {ASN:nn</td>
<td>IP address: nn}`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-vrf)# rd 1:1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td></td>
</tr>
<tr>
<td>`route-target {import</td>
<td>export</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-vrf)#route-target export 1:1</code></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-vrf)#route-target import 1:1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td></td>
</tr>
<tr>
<td><code>address-family ipv6</code></td>
<td>Selects an address family type for a VRF table and enters VRF address family configuration mode. It configures the separate route-target policies for IPv6.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router#(config-vrf)#address-family ipv6</code></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 9</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router#(config-vrf-af)#exit</td>
</tr>
<tr>
<td>Purpose</td>
<td>Exits the address family configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th>address-family ipv4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router#(config-vrf)#address-family ipv4</td>
</tr>
<tr>
<td>Purpose</td>
<td>Selects an address family type for a VRF table and enters VRF address family configuration mode. It configures the separate route-target policies for IPv4.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 11</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router#(config-vrf)#exit</td>
</tr>
<tr>
<td>Purpose</td>
<td>Exits the address family configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 12</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router#(config-vrf)#exit</td>
</tr>
<tr>
<td>Purpose</td>
<td>Exits the VRF configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 13</th>
<th>interface gigabitethernet slot/port</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# interface gigabitethernet 3/1</td>
</tr>
</tbody>
</table>
| Purpose | Enters the interface configuration mode and specifies the Gigabit interface to configure.  
- slot/port—Specifies the location of the interface.  
**Note** This command configures the interface towards the IPv6 network. |

<table>
<thead>
<tr>
<th>Step 14</th>
<th>vrf forwarding vrf name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-if)#vrf forwarding VRF_RED</td>
</tr>
<tr>
<td>Purpose</td>
<td>Associates a VRF instance with an interface or a subinterface.</td>
</tr>
</tbody>
</table>

| Step 15 | ipv6 address {ipv6-address|prefix-length | prefix-name sub-bits | prefix-length} |
|---------|------------------|
| **Example:** | Router (config-if)# ipv6 address 1::2/64 |
| Purpose | Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface. |

<table>
<thead>
<tr>
<th>Step 16</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router (config-if)#exit</td>
</tr>
<tr>
<td>Purpose</td>
<td>Exits interface configuration mode.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 17** | **interface gigabitethernet slot/port**  
**Example:**  
Router(config)# interface gigabitethernet 4/1 |
| Enters the interface configuration mode and specifies the Gigabit interface to configure.  
- *slot/port*—Specifies the location of the interface.  
**Note** This command configures the interface towards the IPv4 network. |
| **Step 18** | **ip address ip-address mask**  
**Example:**  
Router(config-if)# ip address 17.1.1.1 255.255.255.0 |
| Assigns an IP address and subnet mask to the interface. |
| **Step 19** | **ip ospf process-id area area-id**  
**Example:**  
Router(config-if)# ip ospf 2 area 0 |
| Enables the Open Shortest Path First on an interface.  
- *process-id*—Specifies the process ID that ranges from 1 to 65535.  
- *area-id*—Specifies the area ID that ranges from 0 to 4294967295. |
| **Step 20** | **exit**  
**Example:**  
Router(config-if)# exit |
| Exits interface configuration mode. |
| **Step 21** | **interface loopback interface-number**  
**Example:**  
Router(config)# interface Loopback 100 |
| Enters interface configuration mode and names the new loopback interface.  
**Note** This command configures a loopback interface for the tunnel source. |
| **Step 22** | **ip address ip-address mask**  
**Example:**  
Router(config-if)# ip address 66.66.66.66 255.255.255.255 |
| Assigns an IP address and subnet mask to the loopback interface. |
| **Step 23** | **ip ospf process-id area area-id**  
**Example:**  
Router(config-if)# ip ospf 2 area 0 |
| Enables the Open Shortest Path First on an interface.  
- *process-id*—Specifies the process ID that ranges from 1 to 65535.  
- *area-id*—Specifies the area ID that ranges from 0 to 4294967295. |
| **Step 24** | **exit**  
**Example:**  
Router(config-if)# exit |
<p>| Exits interface configuration mode. |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 25**
`interface tunnel`  
tunnel-number | Specifies a tunnel interface and enters the interface configuration mode.  
**Note** This command configures the IPv6 tunneling over IPv4 Transport. |
| **Example:**
Router(config)# interface tunnel 10 | |
| **Step 26**
`vrf forwarding`  
vrf name | Associates a VRF instance with an interface or a subinterface.  
**Note** This command specifies the VRF instance to which the tunnel belongs, that is, the VRF instance used for IPv6 overlay address lookup. |
| **Example:**
Router# (config-if)#vrf forwarding VRF_RED | |
| **Step 27**
`ipv6 address`  
{ipv6-address/prefix-length | prefix-name sub-bits/prefix-length} | Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface. |
| **Example:**
Router(config-if)# ipv6 address 2001:A000:100::1/128 | |
| **Step 28**
`mls 6rd reserve interface`  
gigabitethernet slot/port | Redirects the IPv6 traffic to the IPv4 core facing interface on the ES+ line card. |
| **Example:**
Router(config-if)# mls 6rd reserve interface gig 4/5 | |
| **Step 29**
`tunnel source`  
{ip-address | interface-type interface-number} | Specifies the source interface type and number for the tunnel interface. |
| **Example:**
Router(config-if)# tunnel source loopback 100 | |
| **Step 30**
`tunnel mode ipv6ip`  
{6rd | 6to4 | auto-tunnel | isatap} | Configures a static IPv6 tunnel interface. |
| **Example:**
Router(config-if)# tunnelmode ipv6ip 6rd | |
| **Step 31**
`tunnel 6rd ipv4`  
{(prefix-length length) (suffix-length length)} | Specifies the prefix and suffix length of the IPv4 transport address common to all the 6RD routers in a domain. |
| **Example:**
Router(config-if)# tunnel6rd ipv4 prefix-len 16 suffix-len 8 | |
### Command or Action | Purpose
--- | ---
**Step 32** | **tunnel 6rd prefix**  
*ipv6-prefix/prefix-length*  
*Example:*  
Router(config-if)# tunnel 6rd prefix 2001:A000::/32  
**Specifies the common IPv6 prefix on IPv6 6RD tunnels.**

**Step 33** | **tunnel 6rd br ipv4-address**  
*ipv4-address*  
*Example:*  
Router(config-if)# tunnel 6rd br 60.1.2.1  
**Bypasses security checks on a 6RD customer-edge router.**
- *ipv4-address*—IPv4 address of the border relay (BR) router.

**Step 34** | **exit**  
*Example:*  
Router(config-if)# exit  
**Exits interface configuration mode.**

**Step 35** | **ipv6 route vrf vrf-name**  
*ipv6-prefix/prefix-length*  
*tunnel tunnel number*  
*Example:*  
Router(config)# ipv6 route vrf vrf-red 2001:A000::/32  
Tunnel 10  
**Establishes static routes.**
- *ipv6-prefix*—Specifies the IPv6 network that is the destination of the static route. Can also be a host name when static host routes are configured.
- *prefix-length*—Specifies the length of the IPv6 prefix.

**Step 36** | **ipv6 route vrf vrf-name**  
*ipv6-prefix/prefix-length*  
*tunnel tunnel number*  
*ipv6-address*  
*Example:*  
Router(config)# ipv6 route vrf vrf-red 9000:1000::/64  
Tunnel10 2001:A000:200::1  
**Establishes static routes.**
- *ipv6-address*—The IPv6 address of the next hop that can be used to reach the specified network.

**Step 37** | **end**  
*Example:*  
Router(config)# end  
**Ends the current configuration session.**
Chapter 13 Configuring Layer 3 and Layer 4 Features

DETAILED STEPS (for 6RD Border Relay (BR) Router Configuration)

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Repeat steps 1 through 32 from the 6RD CE configuration, and then continue with these steps:</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>ipv6 route vrf vrf-name ipv6-prefix/prefix-length tunnel tunnel number</td>
</tr>
</tbody>
</table>
| &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n...
Chapter 13 Configuring Layer 3 and Layer 4 Features

VRF aware IPv6 Rapid Deployment (6RD) tunnels

Configuring Layer 3 and Layer 4 Features

6RD Border Relay (BR) Router Configuration

Router# enable
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)# mls ipv6 vrf
Router(config)# vrf definition VRF_RED
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv6
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# exit
Router(config-vrf)# exit
Router(config)# interface gigabitethernet 5/1
Router(config-if)# ipv6 address 9000:1000::/64
Router(config-if)# exit
Router(config)# interface gigabitethernet 4/1
Router(config-if)# ip address 17.1.1.2 255.255.255.0
Router(config-if)# ip ospf 2 area 0
Router(config-if)# exit
Router(config)# interface Loopback 100
Router(config-if)# ip address 60.1.2.1 255.255.255.0
Router(config-if)# ip ospf 2 area 0
Router(config-if)# exit
Router(config)# interface tunnel 10
Router(config-if)# vrf forwarding VRF_RED
Router(config-if)# ipv6 address 2001:A000:100::1/128
Router(config-if)# mls 6rd reserve interface GigabitEthernet4/1
Router(config-if)# tunnel source loopback 100
Router(config-if)# tunnel mode ipv6ip 6rd
Router(config-if)# tunnel 6rd ipv4 prefix-len 16 suffix-len 8
Router(config-if)# tunnel 6rd prefix 2001:A000::/32
Router(config-if)# exit
Router(config)# ipv6 route vrf vrf-red 2001:A000::/32
Router(config)# end

Configuring IPv6 overlay addresses in VRF and IPv4 transport addresses in VRF

The only difference in configuration from the above GRT configuration is the use of the tunnel vrf vrf name command. This command associates a VRF instance to a specific tunnel destination or source.

Complete the following steps to configure IPv6 overlay addresses in VRF, and IPv4 transport addresses in VRF:

6RD customer edge configuration

Step 1 enable
Step 2 configure terminal
Step 3 ipv6 unicast-routing
Step 4 mls ipv6 vrf
Step 5 vrf definition vrf name 1
Step 6 rd {ASN:nn | IP address: nn}
Step 7 route-target [import | export | both] {ASN:nn | IP address: nn}
Step 8 address-family ipv6
Step 9 exit
Step 10 address-family ipv4
Step 11 exit
Step 12 exit
Step 13 vrf definition vrf name 2
Step 14 rd {ASN:nn | IP address: nn}
Step 15 route-target [import | export | both] {ASN:nn | IP address: nn}
Step 16 address-family ipv4
Step 17 exit
Step 18 exit
Step 19 interface gigabitethernet slot/port
Step 20 vrf forwarding vrf name 1
Step 21 ipv6 address {ipv6-address/prefix-length \ prefix-name sub-bits/prefix-length}
Step 22 exit
Step 23 interface gigabitethernet slot/port
Step 24 vrf forwarding vrf name 2
Step 25 ip address ip-address mask
Step 26 ip ospf process-id area area-id
Step 27 exit
Step 28 interface loopback interface-number
Step 29 vrf forwarding vrf name 2
Step 30 ip address ip-address mask
Step 31 ip ospf process-id area area-id
Step 32 exit
Step 33 interface tunnel tunnel-number
Step 34 vrf forwarding vrf name 1
Step 35 ipv6 address {ipv6-address/prefix-length \ prefix-name sub-bits/prefix-length}
Step 36 mls 6rd reserve interface gigabitethernet/ tengigabitethernet
Step 37 tunnel source {ip-address | interface-type interface-number}
Step 38 tunnel mode ipv6ip [6rd | 6to4 | auto-tunnel | isatap]
Step 39 tunnel 6rd ipv4 {prefix-length length} {suffix-length length}
Step 40  tunnel 6rd prefix ipv6-prefix/prefix-length
Step 41  tunnel 6rd br ipv4-address
Step 42  tunnel vrf vrf name 2
Step 43  exit
Step 44  ipv6 route vrf vrf-name ipv6-prefix/prefix-length tunnel tunnel number
Step 45  ipv6 route vrf vrf-name ipv6-prefix/prefix-length tunnel tunnel number ipv6-address
Step 46  end

6RD BR configuration

Step 1  Repeat steps 1 through 40 from the 6RD CE configuration, and then continue with these steps:
Step 2  tunnel vrf vrf name 2
Step 3  exit
Step 4  ipv6 route vrf vrf-name ipv6-prefix/prefix-length tunnel tunnel number
Step 5  end
# DETAILED STEPS (for 6rd CE Configuration)

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>enable</strong>&lt;br&gt;Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>configure terminal</strong>&lt;br&gt;Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>ipv6 unicast-routing</strong>&lt;br&gt;Enables the forwarding of IPv6 unicast datagrams.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# ipv6 unicast-routing</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>mls ipv6 vrf</strong>&lt;br&gt;Enables IPv6 globally in a VRF instance.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# mls ipv6 vrf</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>vrf definition vrf name 1</strong>&lt;br&gt;Configures a VRF instance and enters the VRF configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# vrf definition VRF_RED</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>**rd (ASN:nn</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-vrf)# rd 1:1</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>**route-target [import</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-vrf)#route-target import 1:1&lt;br&gt;Router(config-vrf)#route-target export 1:1&lt;br&gt;Router(config-vrf)#route-target both 1:1</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>address-family ipv6</strong>&lt;br&gt;Selects an address family type for a VRF table and enters VRF address family configuration mode. It configures the separate route-target policies for IPv6.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-vrf)#address-family ipv6</td>
</tr>
</tbody>
</table>
### Chapter 13 Configuring Layer 3 and Layer 4 Features

#### VRF aware IPv6 Rapid Deployment (6RD) tunnels

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong> exit</td>
<td>Exits the address family configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-vrf-af)#exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> address-family ipv4</td>
<td>Selects an address family type for a VRF table and enters VRF address family configuration mode. It configures the separate route-target policies for IPv4.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-vrf)#address-family ipv4</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> exit</td>
<td>Exits the address family configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-vrf-af)#exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> exit</td>
<td>Exits the VRF configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-vrf)#exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> vrf definition vrf name 2</td>
<td>Configures a VRF instance and enters the VRF configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)#vrf definition VRF_GREEN</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong> rd (ASN:nn</td>
<td>Specifies an RD.</td>
</tr>
<tr>
<td>IP address: nn)</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-vrf)# rd 1:1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong> route-target [import</td>
<td>Creates a route-target extended community for a VRF instance. Route target extended community attributes are used to identify a set of sites and VRF instances that can receive routes with a configured route target.</td>
</tr>
<tr>
<td>export</td>
<td>both](ASN:nn</td>
</tr>
<tr>
<td>IP address: nn)</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-vrf)#route-target export 1:1 Router(config-vrf)#route-target import 1:1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong> address-family ipv4</td>
<td>Selects an address family type for a VRF table and enters VRF address family configuration mode. It configures the separate route-target policies for IPv4.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-vrf)#address-family ipv4</td>
<td></td>
</tr>
</tbody>
</table>
### Chapter 13  Configuring Layer 3 and Layer 4 Features

#### VRF aware IPv6 Rapid Deployment (6RD) tunnels

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 17**  
exit  
**Example:** 
`Router(config-vrf-af)#exit` | Exits the address family configuration mode. |
| **Step 18**  
exit  
**Example:** 
`Router(config-vrf)#exit` | Exits the VRF configuration mode. |
| **Step 19**  
`interface gigabitethernet slot/port`  
**Example:** 
`Router(config)# interface gigabitethernet 3/1` | Enters the interface configuration mode and specifies the Gigabit interface to configure.  
- **slot/port**—Specifies the location of the interface.  
**Note** This command configures the interface towards the IPv6 network. |
| **Step 20**  
`vrf forwarding vrf name 1`  
**Example:** 
`Router(config-if)#vrf forwarding VRF_RED` | Associates a VRF instance with an interface or a subinterface. |
| **Step 21**  
`ipv6 address`  
`{ipv6-address|prefix-length | prefix-name sub-bits | prefix-length}`  
**Example:** 
`Router(config-if)# ipv6 address 1::2/64` | Specifies the IPv6 address assigned to the interface, and enables IPv6 processing on the interface. |
| **Step 22**  
exit  
**Example:** 
`Router# (config-if)# exit` | Exits interface configuration mode. |
| **Step 23**  
`interface gigabitethernet slot/port`  
**Example:** 
`Router(config)# interface gigabitethernet 4/5` | Enters the interface configuration mode and specifies the Gigabit interface to configure.  
- **slot/port**—Specifies the location of the interface.  
**Note** This command configures the interface towards the IPv4 network. |
| **Step 24**  
`vrf forwarding vrf name 2`  
**Example:** 
`Router(config-if)#vrf forwarding VRF_GREEN` | Associates a VRF instance with an interface or a subinterface. |
### Command or Action

**Step 25**  
`ip address ip-address mask`  
Assigns an IP address and subnet mask to the interface.

**Example:**  
Router(config-if)#ip address 17.1.1.1 255.255.255.0

**Step 26**  
`ip ospf process-id area area-id`  
 Enables the Open Shortest Path First on an interface.  
- `process-id`—Specifies the process ID that ranges from 1 to 65535.  
- `area-id`—Specifies the area ID that ranges from 0 to 4294967295.

**Example:**  
Router(config-if)#ip ospf 2 area 0

**Step 27**  
`exit`  
Exits interface configuration mode.

**Example:**  
Router(config-if)# exit

**Step 28**  
`interface loopback interface-number`  
Enters interface configuration mode and names the new loopback interface.  
**Note** This command configures a loopback interface for the tunnel source.

**Example:**  
Router(config)# interface Loopback 100

**Step 29**  
`vrf forwarding vrf name`  
Associates a VRF instance with an interface or a subinterface.

**Example:**  
Router(config-if)#vrf forwarding VRF_GREEN

**Step 30**  
`ip address ip-address`  
Assigns an IP address and subnet mask to the loopback interface.

**Example:**  
Router(config-if)#ip address 60.1.1.1 255.255.255.255

**Step 31**  
`ip ospf process-id area area-id`  
 Enables the Open Shortest Path First on an interface.  
- `process-id`—Specifies the process ID that ranges from 1 to 65535.  
- `area-id`—Specifies the area ID that ranges from 0 to 4294967295.

**Example:**  
Router(config-if)#ip ospf 2 area 0

**Step 32**  
`exit`  
Exits interface configuration mode.

**Example:**  
Router(config-if)# exit
### Command or Action

**Step 33**

```
interface tunnel
tunnel-number
```

**Example:**

```
Router(config)# interface
tunnel 10
```

- **Purpose:** Specifies a tunnel interface and enters the interface configuration mode.
- **Note:** This command configures the IPv6 tunneling over IPv4 Transport.

**Step 34**

```
vrf forwarding vrf name 1
```

**Example:**

```
Router(config-if)# vrf
forwarding VRF_RED
```

- **Purpose:** Associates a VRF instance with an interface or a subinterface.
- **Note:** This command specifies the VRF instance to which the tunnel belongs, that is, the VRF instance used for IPv6 overlay address lookup.

**Step 35**

```
ipv6 address {ipv6-address|prefix-length
|prefix-name sub-bits|prefix-length}
```

**Example:**

```
Router(config-if)# ipv6
address 2001:A000:100::1/128
```

- **Purpose:** Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface.

**Step 36**

```
mls 6rd reserve interface gigabitethernet slot/port
```

**Example:**

```
Router(config-if)# mls 6rd
reserve interface gig 4/5
```

- **Purpose:** Redirects the IPv6 traffic to the IPv4 core facing interface on the ES+ line card.

**Step 37**

```
tunnel source {ip-address | interface-type
|interface-number}
```

**Example:**

```
Router(config-if)# tunnel
source loopback 100
```

- **Purpose:** Specifies the source interface type and number for the tunnel interface.

**Step 38**

```
tunnel mode ipv6ip [6rd | 6to4 | auto-tunnel | isatap]
```

**Example:**

```
Router(config-if)# tunnel
mode ipv6ip 6rd
```

- **Purpose:** Configures a static IPv6 tunnel interface.

**Step 39**

```
tunnel 6rd ipv4 {prefix-length | length}
(suffix-length | length)
```

**Example:**

```
Router(config-if)# tunnel6rd ipv4 prefix-len
16 suffix-len 8
```

- **Purpose:** Specifies the prefix and suffix length of the IPv4 transport address common to all the 6RD routers in a domain.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 40 tunnel 6rd prefix ipv6-prefix/prefix-length</td>
<td>Specifies the common IPv6 prefix on IPv6 6RD tunnels.</td>
</tr>
<tr>
<td>Example: Router(config-if)# tunnel 6rd prefix 2001:A000::/32</td>
<td></td>
</tr>
<tr>
<td>Step 41 tunnel 6rd br ipv4-address</td>
<td>Bypasses security checks on a 6RD customer-edge router.</td>
</tr>
<tr>
<td>Example: Router(config-if)# tunnel 6rd br 60.1.2.1</td>
<td>• ipv4-address—IPv4 address of the border relay (BR) router.</td>
</tr>
<tr>
<td>Step 42 tunnel vrf vrf name 2</td>
<td>Configures a VRF instance with a specific tunnel destination, interface or a subinterface.</td>
</tr>
<tr>
<td>Example: Router(config-if)# tunnel vrf VRF_GREEN</td>
<td>Note This command specifies the VRF instance used for the tunnel IPv4 transport address lookup.</td>
</tr>
<tr>
<td>Step 43 exit</td>
<td>Exits interface configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 44 ipv6 route vrf vrf-name ipv6-prefix/prefix-length tunnel tunnel number</td>
<td>Establishes static routes.</td>
</tr>
<tr>
<td>Example: Router(config)# ipv6 route vrf vrf-red 2001:A000::/32 Tunnel 10</td>
<td>• ipv6-prefix—Specifies the IPv6 network that is the destination of the static route. Can also be a host name when static host routes are configured.</td>
</tr>
<tr>
<td></td>
<td>• prefix-length—Specifies the length of the IPv6 prefix.</td>
</tr>
<tr>
<td>Step 45 ipv6 route vrf vrf-name ipv6-prefix/prefix-length tunnel tunnel number ipv6-address</td>
<td>Establishes static routes.</td>
</tr>
<tr>
<td>Example: Router(config)#ipv6 route vrf vrf-red 9000:1000::/64 Tunnel10 2001:A000:200::1</td>
<td>• ipv6-address—The IPv6 address of the next hop that can be used to reach the specified network.</td>
</tr>
<tr>
<td>Step 46 end</td>
<td>Ends the current configuration session.</td>
</tr>
<tr>
<td>Example: Router(config)# end</td>
<td></td>
</tr>
</tbody>
</table>
**DETAILED STEPS (for 6RD BR Configuration)**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Repeat steps 1 through 40 from the 6RD CE configuration, and then continue with these steps:</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>tunnel vrf vrf name 2</code></td>
<td>Configures a VRF instance with a specific tunnel destination, interface or a subinterface.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-if)# tunnel vrf VRF_GREEN</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>This command specifies the VRF instance used for tunnel IPv4 transport address lookup.</td>
</tr>
<tr>
<td>3</td>
<td><code>exit</code></td>
<td>Exits interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-if)# exit</code></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>ipv6 route vrf vrf-name ipv6-prefix/prefix-length tunnel tunnel number</code></td>
<td>Establishes static routes.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config)# ipv6 route vrf vrf-red 2001:A000::/32 Tunnel 10</code></td>
<td></td>
</tr>
</tbody>
</table>
|      | **Note** | - `ipv6-prefix`—Specifies the IPv6 network that is the destination of the static route. Can also be a host name when static host routes are configured.  
- `prefix-length`—Specifies the length of the IPv6 prefix. |
| 5    | `end` | Ends the current configuration session. |
|      | **Example:** | |
|      | `Router(config)# end` | |

**Configuration Example**

This example shows how to configure IPv6 overlay addresses in VRF, and IPv4 transport addresses in VRF:

**6RD customer edge configuration**

```
Router# enable
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)# mls ipv6 vrf
Router(config)# vrf definition VRF_RED
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv6
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# exit
Router(config-vrf)# exit
Router(config)# vrf definition VRF_GREEN
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# exit
Router(config-vrf)# exit
```
Chapter 13  Configuring Layer 3 and Layer 4 Features

VRF aware IPv6 Rapid Deployment (6RD) tunnels

Router(config)# interface gigabitethernet 3/1
Router(config-if)# vrf forwarding VRF_RED
Router(config-if)# ipv6 address 1::2/64
Router(config-if)# exit
Router(config)# interface gigabitethernet 4/5
Router(config-if)# vrf forwarding VRF_GREEN
Router(config-if)# ip address 17.1.1.1 255.255.255.0
Router(config-if)# ip ospf 2 area 0
Router(config-if)# exit
Router(config)# interface Loopback 100
Router(config-if)# vrf forwarding VRF_GREEN
Router(config-if)# ip address 60.1.1.1 255.255.255.0
Router(config-if)# ip ospf 2 area 0
Router(config-if)# exit
Router(config)# interface tunnel 10
Router(config-if)# vrf forwarding VRF_RED
Router(config-if)# ipv6 address 2001:A000:100::1/128
Router(config-if)# mls 6rd reserve interface GigabitEthernet4/5
Router(config-if)# tunnel source loopback 100
Router(config-if)# tunnel mode ipv6ip 6rd
Router(config-if)# tunnel 6rd ipv4 prefix-len 16 suffix-len 8
Router(config-if)# tunnel 6rd prefix 2001:A000::/32
Router(config-if)# tunnel 6rd br 60.1.2.1
Router(config-if)# tunnel vrf VRF_GREEN
Router(config-if)# exit
Router(config)#ipv6 route vrf vrf-red 2001:A000::/32 Tunnel10
Router(config)#ipv6 route vrf vrf-red 9000:1000::/64 Tunnel10 2001:A000:200::1
Router(config)# end

6RD BR config

Router# enable
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)# mls ipv6 vrf
Router(config)# vrf definition VRF_RED
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv6
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# exit
Router(config-vrf)# exit
Router(config)# vrf definition VRF_GREEN
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# exit
Router(config-vrf)# interface gigabitethernet 5/1
Router(config-if)# vrf forwarding VRF_RED
Router(config-if)# ipv6 address 9000:1000::/64
Router(config-if)# exit
Router(config)# interface gigabitethernet 4/1
Router(config-if)# vrf forwarding VRF_GREEN
Router(config-if)# ip address 17.1.1.2 255.255.255.0
Router(config-if)# ip ospf 2 area 0
Router(config-if)# exit
Router(config)# interface Loopback 100
Router(config-if)# vrf forwarding VRF_GREEN
Router(config-if)# ip address 60.1.2.1 255.255.255.0
Chapter 13  Configuring Layer 3 and Layer 4 Features

Verifying the Configuration

Use these commands to verify the configuration of VRF aware 6RD tunnels on c7600:

- show platform npc ipv6_6rd egress-table vlan tunnel-vlan

Router# show platform npc ipv6_6rd egress-table vlan 1013
IPV6 6rd endpoint data for tepi_idx 0
dip 0.0.0.0 dmac-smac 0023.0417.b1c0-0023.0417.b1c0
Phy vlan 1024 Active 1 Tunnel Vlan 1013

- show platform npc ipv6_6rd egress-table vlan tunnel-vlan detail

Router# show platform npc ipv6_6rd egress-table vlan 1013 detail
IPV6_6RD egress table entry
eg_entry->match_cond = 1
eg_entry->ent_valid = 1
eg_entry->static_route = 0
eg_entry->src_ip = 10.1.4.1
eg_entry->v4_add_mask = 0.0.255.0
eg_entry->v4_pref_suff = 10.1.0.1
eg_entry->v4_sp_pref_byte_off = 4
eg_entry->v4_sp_pref_bit_off = 0
eg_entry->v4_pref_in_bits = 16
eg_entry->is_tun_ep = 0
dmac 0023.0417.B1C0
smac 0023.0417.B1C0
eg_entry->eg_stats_id = 312020 0x0004C2D4
eg_entry->phy_vlan = 1024

value: 04 00 40 03 0a 01 04 01 00 00 00 ff 00 0a 01 00 01 ..@............
value: 00 23 04 17 00 23 04 17 b1 c0 b1 c0 10 04 c2 d4 .#...#..1010..BT

- show platform npc 6rd tcam vlan 1061

Router# sh plat npc 6rd tcam vlan 1061
TCAM entry for tunnel vlan 1061 on np 0
VMR-Handle : 132
Source IP : 100.0.32.1 Mask : 00000000
VRF id : 0 Mask : 0000
Feature id : 2 Mask : 00
Result Decode :
Vlan : 1061
Statistics ID : 0x53DBA

Raw output :
Key : 64 00 20 01 00 00 00 00 02 00 00 00 00 00 00 00 04 25 53 20
VRF aware IPv6 Rapid Deployment (6RD) tunnels

Chapter 13  Configuring Layer 3 and Layer 4 Features

VRF aware IPv6 Rapid Deployment (6RD) tunnels

TCAM entry for tunnel vlan 1061 on np 1
VMR-Handle : 132

Source IP : 100.0.32.1 Mask : 00000000
VRF id : 0 Mask : 00
Feature id : 2 Mask : 00
Result Decode :
Vlan : 1061
Statistics ID : 0x53DBA

Raw output :
Key : 64 00 20 01 00 00 00 00 02 00 00 00 00 00 00 00 04 25 53 20
Mask : 00 00 00 00 FF FF FF FF 00 FF 00 00 FF FF FF FF FF FF A6 B0
Result : 04 25 01 03 00 05 3D BA

- show platform npc 6rd xlif vlan 1061

Router# show platform npc 6rd xlif vlan 1061
Eg xlif id (1061 + 32000) = (33061) tunnel_vlan : 1061

Egress XLIF table fields

Feature common enable: 0x1
Feature enable: 0x1
Feature bits: 0x04
Control common bits: 0x00
Control feature bits: 0x00
Control rewrite opcode: 0x00
Port: 0x3
Match cond 0x1
Entry valid: 0x1
Optimal Path en : 0x0
Dbus VLAN: 1018
QoS policy ID: 0
ACL ID: 0
Statistics ID: 0
Inner rewrite VLAN: 0
Outer rewrite VLAN: 0
QoS flow ID: 0
IP Session en : 0
Feature data 0 0x80830008
Intf etype: 0x00000000
Multicast enable: 0x00000000
Post Filter Opcode 0x00000000
Pre Filter Opcode 0x00000000
Pre Tag Outer 0x00000000
Pre Tag Inner 0x00000083
Post Filter Vlan high 0x00000000
Post Filter Vlan low 0x00000000
Post Filter Vlan outer 0x00000000
EVC - MST: 0x0
EVC etype 0x03FA
CFM MEP Level 0x00000000
CFM MIP Level 0x00000000
CFM disable 0x0
MIP filtering 0x0
block_data: 0x0
block_l2bpdu: 0x0
sacl: 0x0
MVPNV6 decap Vlanv4: 31
MVPNV6 decap Vlanv6: 1018
VRF aware IPv6 Rapid Deployment (6RD) tunnels

Chapter 13  Configuring Layer 3 and Layer 4 Features

VRF aware IPv6 Rapid Deployment (6RD) tunnels

sacl index: 0x0000
sacl statid: 0x10000
Span Enable: 0x0

Eg xlif id (1018 + 32000) phy_vlan : (1018)

Egress XLIF table fields

- Feature common enable: 0x1
- Feature enable: 0x1
- Feature bits: 0x01
- Control common bits: 0x00
- Control feature bits: 0x00
- Control rewrite opcode: 0x00
- Port: 0x3
- Match cond: 0x1
- Entry valid: 0x1
- Optimal Path en: 0x1
- Dbus VLAN: 1018
- QoS policy ID: 0
- ACL ID: 0
- Statistics ID: 0
- Inner rewrite VLAN: 0
- Outer rewrite VLAN: 0
- QoS flow ID: 0
- IP Session en: 0
- Feature data 0: 0x00830008
- Intf etype: 0x00008100
- Multicast enable: 0x00000000
- Post Filter Opcode: 0x00000008
- Pre Filter Opcode: 0x00000000
- Pre Tag Outer: 0x00000008
- Pre Tag Inner: 0x00000083
- Post Filter Vlan high: 0x00000000
- Post Filter Vlan low: 0x00000100
- Post Filter Vlan outer: 0x00000000
- EVC - MST: 0x0
- EVC etype: 0x0000
- CFM MEP Level: 0x00000000
- CFM MIP Level: 0x00000000
- CFM disable: 0x0
- MIP filtering: 0x0
- block_data: 0x0
- block_l2bpdu: 0x0
- sacl: 0x0
- MVPNV6 decap Vlanv4: 0
- MVPNV6 decap Vlanv6: 0
- sacl index: 0x0000
- sacl statid: 0x000000
- Span Enable: 0x0

- show platform npc 6rd tunnel 34

Router# show platform npc 6rd tunnel 34
Tunnel34 is up, line protocol is up
Hardware is Tunnel
MTU 0 bytes, BW 1000000 Kbit/sec, DLY 0 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive set (10 sec)
Tunnel source 100.0.34.1 (Loopback34), destination 0.0.0.0
Tunnel protocol/transport IPv6 6RD, key disabled, sequencing disabled
Checksumming of packets disabled, vip tunneling disabled
Last input never, output never, output hang never
Last clearing of "show interface" counters 1d02h
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
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 no buffer
Received 0 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 unknown protocol drops
0 output buffer failures, 0 output buffers swapped out

IPV6 6RD tunnel information on CP: Tunnel34, interface vlan = 1063
if_num = 1750, phy_vlan = 1018, tun_vlan = 1063, id =33, inuse = 1, active = 1 error = 0
sip 100.0.34.1, ep count 2

TEPI indices of remote endpoint associated with the IPV6 6RD Tunnel interface are:
35

Troubleshooting

For troubleshooting information, contact Cisco Technical Assistance Center (TAC) at:
VRF aware IPv6 Tunnels over IPv4 Transport

The current IPv6 tunneling feature on c7600 does not support Virtual Routing and Forwarding (VRF) awareness. The forwarding table lookups for IPv6 overlay addresses and IPv4 transport addresses are performed in the global routing tables. This feature extends the tunneling support for IPv6 overlay addresses in VRF.

These scenarios explain the VRF aware IPv6 tunnel function:
- IPv6 overlay address in VRF and IPv4 transport address in Global routing table (RT).
- IPv6 overlay address in VRF and IPv4 transport address in VRF.

Figure 13-5 illustrates the topology for the IPv6 overlay address in VRF, and the IPv4 transport address in VRF.

Figure 13-5 Topology for VRF aware IPv6 Tunnel

The VRF Aware IPv6 over IPv4 Tunnel can have any line card towards the core facing side.

Restrictions for VRF aware IPv6 tunnels

Following restrictions apply to the VRF aware IPv6 tunnels feature:
This feature supports the IPv6IP and 6to4 tunnels mode.
Due to EARL limitation, the same source tunnels across VRFs are not supported.
The tunnel source and the tunnel destination should be in the same VRF instance.
The tunnel IPv4 transport addresses and the physical interface where the tunnel traffic exits, should be in the same VRF instance.
The incoming IPv6 interface and the tunnel should be in the same VRF instance.
This feature does not support IPv6IP auto-tunnels and ISATAP.

Configuring VRF aware IPv6 tunnel

The following sections describe how to configure VRF aware IPv6 tunnel on c7600:
- Configure IPv6 overlay addresses in VRF and IPv4 transport addresses in Global RT, page 13-63
- Configure IPv6 overlay addresses in VRF and IPv4 transport addresses in VRF, page 13-69

Configure IPv6 overlay addresses in VRF and IPv4 transport addresses in Global RT

Complete the following steps to configure IPv6 overlay addresses in VRF and IPv4 transport addresses in Global RT:

**SUMMARY STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>enable</td>
</tr>
<tr>
<td>2</td>
<td>configure terminal</td>
</tr>
<tr>
<td>3</td>
<td>ipv6 unicast-routing</td>
</tr>
<tr>
<td>4</td>
<td>mls ipv6 vrf</td>
</tr>
<tr>
<td>5</td>
<td>vrf definition vrf name</td>
</tr>
<tr>
<td>6</td>
<td>rd {ASN:nn</td>
</tr>
<tr>
<td>7</td>
<td>route-target [import</td>
</tr>
<tr>
<td>8</td>
<td>address-family ipv6</td>
</tr>
<tr>
<td>9</td>
<td>exit</td>
</tr>
<tr>
<td>10</td>
<td>address-family ipv4</td>
</tr>
<tr>
<td>11</td>
<td>exit</td>
</tr>
<tr>
<td>12</td>
<td>exit</td>
</tr>
<tr>
<td>13</td>
<td>interface gigabitethernet slot/port</td>
</tr>
<tr>
<td>14</td>
<td>vrf forwarding vrf name</td>
</tr>
<tr>
<td>15</td>
<td>ipv6 address {ipv6-address/prefix-length</td>
</tr>
<tr>
<td>16</td>
<td>exit</td>
</tr>
<tr>
<td>17</td>
<td>interface gigabitethernet slot/port</td>
</tr>
<tr>
<td>18</td>
<td>ip address ip-address</td>
</tr>
<tr>
<td>19</td>
<td>exit</td>
</tr>
</tbody>
</table>
Step 20 interface loopback interface-number
Step 21 ip address ip-address
Step 22 exit
Step 23 interface tunnel tunnel-number
Step 24 vrf forwarding vrf name
Step 25 ipv6 address {ipv6-address/prefix-length \ prefix-name sub-bits/prefix-length}
Step 26 tunnel source {ip-address \ interface-type interface-number}
Step 27 tunnel destination {hostname \ ip-address \ ipv6-address}
Step 28 tunnel mode ipv6ip
Step 29 end
## Chapter 13  Configuring Layer 3 and Layer 4 Features

### VRF aware IPv6 Tunnels over IPv4 Transport

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# enable</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv6 unicast-routing</td>
<td>Enables the forwarding of IPv6 unicast datagrams.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# ipv6 unicast-routing</td>
</tr>
<tr>
<td><strong>Step 4</strong> mls ipv6 vrf</td>
<td>Enables IPv6 globally in a VRF instance.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# mls ipv6 vrf</td>
</tr>
<tr>
<td><strong>Step 5</strong> vrf definition vrf name</td>
<td>Configures a VRF instance and enters the VRF configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# vrf definition VRF_RED</td>
</tr>
<tr>
<td><strong>Step 6</strong> rd (ASN:nn</td>
<td>IP address: nn)</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-vrf)# rd 1:1</td>
</tr>
<tr>
<td><strong>Step 7</strong> route-target [import</td>
<td>export</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-vrf)#route-target export 1:1</td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf)#route-target import 1:1</td>
</tr>
<tr>
<td><strong>Step 8</strong> address-family ipv6</td>
<td>Selects an address family type for a VRF table and enters VRF address family configuration mode. This command configures the separate route-target policies for IPv6.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router#(config-vrf)#address-family ipv6</td>
</tr>
</tbody>
</table>
### VRF aware IPv6 Tunnels over IPv4 Transport

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><strong>exit</strong></td>
<td>Exits the address family configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router#(config-vrf-af)#exit</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><strong>address-family ipv4</strong></td>
<td>Selects an address family type for a VRF table and enters VRF address family configuration mode. This command configures the separate route-target policies for IPv4.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router#(config-vrf)#address-family ipv4</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><strong>exit</strong></td>
<td>Exits the address family configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router#(config-vrf-af)#exit</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td><strong>exit</strong></td>
<td>Exits the VRF configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router#(config-vrf)#exit</td>
<td></td>
</tr>
</tbody>
</table>
| 13    | **interface gigabitethernet slot/port** | Enters the interface configuration mode and specifies the Gigabit interface to configure.  
- *slot/port*—Specifies the location of the interface.  
**Note** This command configures the interface towards the IPv6 network. |
|       | **Example:**     |         |
|       | Router(config)# interface gigabitethernet 3/1 |         |
| 14    | **vrf forwarding vrf name** | Associates a VRF instance with an interface or a subinterface. |
|       | **Example:**     |         |
|       | Router(config-if)#vrf forwarding VRF_RED |         |
| 15    | **ipv6 address** | Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface.  
- *(ipv6-address|prefix-length | prefix-name sub-bits | prefix-length)* |
|       | **Example:**     |         |
|       | Router (config-if)# ipv6 address 1::2/64 |         |
| 16    | **exit**          | Exits interface configuration mode. |
|       | **Example:**     |         |
|       | Router (config-if)#exit |         |
### VRF aware IPv6 Tunnels over IPv4 Transport

#### Step 17
**Command or Action:**
```text
interface gigabitethernet
slot/port
```
**Purpose:**
Enters the interface configuration mode and specifies the Gigabit interface to configure.
- `slot/port`—Specifies the location of the interface.
**Example:**
```
Router(config)# interface gigabitethernet 4/1
```
**Note**
This command configures the interface towards the IPv4 network.

#### Step 18
**Command or Action:**
```text
ip address ip-address
```
**Purpose:**
Assigns an IP address and subnet mask to the interface.
**Example:**
```
Router(config-if)# ip address 10.1.1.1 255.255.255.0
```

#### Step 19
**Command or Action:**
```text
exit
```
**Purpose:**
Exits interface configuration mode.
**Example:**
```
Router(config-if)# exit
```

#### Step 20
**Command or Action:**
```text
interface loopback
interface-number
```
**Purpose:**
Enters interface configuration mode and names the new loopback interface.
**Note**
This command configures a loopback interface for the tunnel source
**Example:**
```
Router(config)# interface Loopback 666
```

#### Step 21
**Command or Action:**
```text
ip address ip-address
```
**Purpose:**
Assigns an IP address and subnet mask to the loopback interface.
**Example:**
```
Router(config-if)# ip address 66.66.66.66 255.255.255.255
```

#### Step 22
**Command or Action:**
```text
exit
```
**Purpose:**
Exits interface configuration mode.
**Example:**
```
Router(config-if)# exit
```

#### Step 23
**Command or Action:**
```text
interface tunnel
tunnel-number
```
**Purpose:**
Specifies a tunnel interface and enters the interface configuration mode.
**Note**
This command configures the IPv6 tunneling over IPv4 Transport.
**Example:**
```
Router(config)# interface tunnel 666
```

#### Step 24
**Command or Action:**
```text
vrf forwarding vrf name
```
**Purpose:**
Associates a VRF instance with an interface or a subinterface.
**Note**
This command specifies the VRF instance to which the tunnel belongs, that is, the VRF instance used for IPv6 overlay address lookup.
**Example:**
```
Router# (config-if)# vrf forwarding VRF_RED
```
Chapter 13  Configuring Layer 3 and Layer 4 Features

Step 25  
**Command or Action:**  
`ipv6 address`  
(ipv6-address/prefix-length | prefix-name  
sub-bits/prefix-length)  

**Purpose:** Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface.

**Example:**  
Router(config-if)# ipv6 address 3::1/120

Step 26  
**Command or Action:**  
`tunnel source`  
(ip-address | interface-type  
interface-number)  

**Purpose:** Specifies the source interface type and number for the tunnel interface.

**Example:**  
Router(config-if)# tunnel source loopback 666

Step 27  
**Command or Action:**  
`tunnel destination`  
(host-name | ip-address |  
ipv6-address)  

**Purpose:** Specifies the destination address for a tunnel interface.

**Example:**  
Router(config-if)# tunnel destination 10.66.66.1

Step 28  
**Command or Action:**  
`tunnel mode ipv6ip [6rd |  
6to4 | auto-tunnel |  
isatap]`  

**Purpose:** Configures a static IPv6 tunnel interface.

**Example:**  
Router(config-if)# tunnel mode ipv6ip

Step 29  
**Command or Action:**  
`end`  

**Purpose:** Ends the current configuration session.

**Example:**  
Router(config-if)# end

### Configuration Example

This example shows how to configure the IPv6 overlay addresses in VRF, and the IPv4 transport addresses in the Global Routing Table:

`Router# enable`  
`Router# configure terminal`  
`Router(config)# ipv6 unicast-routing`  
`Router(config)# mls ipv6 vrf`  
`Router(config)# vrf definition VRF_RED`  
`Router(config-vrf)# rd 1:1`  
`Router(config-vrf)# route-target export 1:1`  
`Router(config-vrf)# route-target import 1:1`  
`Router(config-vrf)# address-family ipv6`  
`Router(config-vrf-af)# exit`  
`Router(config-vrf)# address-family ipv4`  
`Router(config-vrf)# (config-vrf-af)# exit`  
`Router(config-vrf)# exit`  
`Router(config)# interface gigabitethernet 3/1`
Configure IPv6 overlay addresses in VRF and IPv4 transport addresses in VRF

Complete the following steps to configure IPv6 overlay addresses in VRF, and IPv4 transport addresses in VRF:

**SUMMARY STEPS**

1. enable
2. configure terminal
3. ipv6 unicast-routing
4. mls ipv6 vrf
5. vrf definition vrf name 1
6. rd \{ASN:nn | IP address: nn\}
7. route-target [import | export | both] [ASN:nn | IP address: nn]
8. address-family ipv6
9. exit
10. address-family ipv4
11. exit
12. exit
13. vrf definition vrf name 2
14. rd \{ASN:nn | IP address: nn\}
15. route-target [import | export | both] [ASN:nn | IP address: nn]
16. address-family ipv4
17. exit
18. exit
19. interface gigabitethernet slot/port
20. vrf forwarding vrf name 1
21. ipv6 address \{ipv6-address/prefix-length | prefix-name sub-bits/prefix-length\}
22. exit
Step 23  interface gigabitethernet slot/port
Step 24  vrf forwarding vrf name 2
Step 25  ip address ip-address
Step 26  exit
Step 27  interface loopback interface-number
Step 28  vrf forwarding vrf name 2
Step 29  ip address ip-address
Step 30  exit
Step 31  interface tunnel tunnel-number
Step 32  vrf forwarding vrf name 1
Step 33  ipv6 address {ipv6-address/prefix-length | prefix-name sub-bits/prefix-length}
Step 34  tunnel source {ip-address | interface-type interface-number}
Step 35  tunnel destination {hostname | ip-address | ipv6-address}
Step 36  tunnel mode ipv6ip
Step 37  tunnel vrf vrf name 2
Step 38  end
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv6 unicast-routing</td>
<td>Enables the forwarding of IPv6 unicast datagrams.</td>
</tr>
<tr>
<td><strong>Step 4</strong> mls ipv6 vrf</td>
<td>Enables IPv6 globally in a VRF instance.</td>
</tr>
<tr>
<td><strong>Step 5</strong> vrf definition vrf name 1</td>
<td>Configures a VRF instance and enters the VRF configuration mode.</td>
</tr>
<tr>
<td><strong>Step 6</strong> rd {ASN:nn</td>
<td>IP address: nn}</td>
</tr>
<tr>
<td><strong>Step 7</strong> route-target [import</td>
<td>export</td>
</tr>
<tr>
<td><strong>Step 8</strong> address-family ipv6</td>
<td>Select an address family type for a VRF table and enters VRF address family configuration mode. This command configures the separate route-target policies for IPv6.</td>
</tr>
</tbody>
</table>

**Example:**

- `Router# enable`
- `Router(config)# ipv6 unicast-routing`
- `Router(config)# mls ipv6 vrf`
- `Router(config)# vrf definition VRF_RED`
- `Router(config-vrf)# rd 1:1`
- `Router(config-vrf)# route-target import 1:1`
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>exit</td>
<td>Exits the address family configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf-af)#exit</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>address-family ipv4</td>
<td>Selects an address family type for a VRF table and enters VRF address family configuration mode. This command configures the separate route-target policies for IPv4.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf)#address-family ipv4</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>exit</td>
<td>Exits the address family configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf-af)#exit</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>exit</td>
<td>Exits the VRF configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf)#exit</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>vrf definition vrf name 2</td>
<td>Configures a VRF instance and enters the VRF configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config)#vrf definition VRF_GREEN</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>rd (ASN:nn</td>
<td>IP address: nn)</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf)#rd 1:1</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>route-target [import</td>
<td>export</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf)#route-target export 1:1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf)#route-target import 1:1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf)#route-target import 1:1</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>address-family ipv4</td>
<td>Selects an address family type for a VRF table and enters VRF address family configuration mode. This command configures the separate route-target policies for IPv4.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf)#address-family ipv4</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>17</td>
<td><code>exit</code></td>
<td>Exits the address family configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> &lt;br&gt;Router (config-vrf-af)#exit</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td><code>exit</code></td>
<td>Exits the VRF configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> &lt;br&gt;Router(config-vrf)#exit</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td><code>interface gigabitethernet slot/port</code></td>
<td>Enters the interface configuration mode and specifies the Gigabit interface to configure.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> &lt;br&gt;Router(config)# interface gigabitethernet 3/1</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>This command configures the interface towards the IPv6 network.</td>
</tr>
<tr>
<td>20</td>
<td><code>vrf forwarding vrf name 1</code></td>
<td>Associates a VRF instance with an interface or a subinterface.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> &lt;br&gt;Router(config-if)#vrf forwarding VRF_RED</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>`ipv6 address {ipv6-address</td>
<td>prefix-length</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> &lt;br&gt;Router(config-if)# ipv6 address 1::2/64</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td><code>exit</code></td>
<td>Exits interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> &lt;br&gt;Router# (config-if)# exit</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td><code>interface gigabitethernet slot/port</code></td>
<td>Enters the interface configuration mode and specifies the Gigabit interface to configure.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> &lt;br&gt;Router(config)# interface gigabitethernet 4/1</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>This command configures the interface towards the IPv4 network.</td>
</tr>
<tr>
<td>24</td>
<td><code>vrf forwarding vrf name 2</code></td>
<td>Associates a VRF instance with an interface or a subinterface.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> &lt;br&gt;Router(config-if)#vrf forwarding VRF_GREEN</td>
<td></td>
</tr>
</tbody>
</table>
### Step 25
**Command or Action**

`ip address ip-address`

**Purpose**

Assigns an IP address and subnet mask to the interface.

**Example:**

Router(config-if)#ip address 10.1.1.1
255.255.255.0

---

### Step 26
**Command or Action**

`exit`

**Purpose**

Exits interface configuration mode.

**Example:**

Router(config-if)# exit

---

### Step 27
**Command or Action**

`interface loopback interface-number`

**Purpose**

Enters interface configuration mode and names the new loopback interface.

**Example:**

Router(config)# interface Loopback 666

**Note**

This command configures a loopback interface for the tunnel source.

---

### Step 28
**Command or Action**

`vrf forwarding vrf name 2`

**Purpose**

Associates a VRF instance with an interface or a subinterface.

**Example:**

Router(config-if)#vrf forwarding VRF_GREEN

---

### Step 29
**Command or Action**

`ip address ip-address`

**Purpose**

Assigns an IP address and subnet mask to the loopback interface.

**Example:**

Router(config-if)#ip address 66.66.66.66
255.255.255.255

---

### Step 30
**Command or Action**

`exit`

**Purpose**

Exits interface configuration mode.

**Example:**

Router(config-if)# exit

---

### Step 31
**Command or Action**

`interface tunnel tunnel-number`

**Purpose**

Specifies a tunnel interface and enters the interface configuration mode.

**Note**

This command configures the IPv6 tunneling over IPv4 Transport.

**Example:**

Router(config)# interface tunnel 666

---

### Step 32
**Command or Action**

`vrf forwarding vrf name 1`

**Purpose**

Associates a VRF instance with an interface or a subinterface.

**Note**

This command specifies the VRF instance to which the tunnel belongs, that is, the VRF instance used for IPv6 overlay address lookup.

**Example:**

Router(config-if)#vrf forwarding VRF_RED
This example shows how to configure the IPv6 overlay addresses in VRF, and the IPv4 transport addresses in VRF:

```
Router# enable
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)# mls ipv6 vrf
Router(config)# vrf definition VRF_RED
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv6
Router(config-vrf-af)# exit
```

**Command or Action** | **Purpose**
---|---
Step 33 `ipv6 address {ipv6-address/prefix-length | prefix-name sub-bits/prefix-length}` | Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface.

Example:
```
Router(config-if)# ipv6 address 3::1/120
```

Step 34 `tunnel source {ip-address | interface-type interface-number}` | Specifies the source interface type and number for the tunnel interface.

Example:
```
Router(config-if)# tunnel source loopback 666
```

Step 35 `tunnel destination {host-name | ip-address | ipv6-address}` | Specifies the destination address for a tunnel interface.

Example:
```
Router(config-if)# tunnel destination 10.66.66.1
```

Step 36 `tunnel mode ipv6ip` | Configures a static IPv6 tunnel interface.

Example:
```
Router(config-if)# tunnel mode ipv6ip
```

Step 37 `tunnel vrf vrf name 2` | Configures a VRF instance with a specific tunnel destination, interface or a subinterface.

Example:
```
Router(config-if)# tunnel vrf VRF_GREEN
```

**Note** This command specifies the VRF instance used for tunnel IPv4 transport address lookup, that is, the tunnel source and the tunnel destination.

Step 38 `end` | Ends the current configuration session.

Example:
```
Router(config-if)# end
```

**Configuration Example**

This example shows how to configure the IPv6 overlay addresses in VRF, and the IPv4 transport addresses in VRF:

```
Router# enable
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)# mls ipv6 vrf
Router(config)# vrf definition VRF_RED
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv6
Router(config-vrf-af)# exit
```
Verifying the Configuration

Use these commands to verify the configuration of VRF aware IPv6 tunnel on c7600:

Router# show vrf vrf-red
Name          Default RD  Protocols  Interfaces
vrf-red        100:1    ipv4,ipv6  Tu666

Router# show interface tunnel 666
Tunnel666 is up, line protocol is up
Hardware is Tunnel
Internet address is 80.1.1.1/24
MTU 17916 bytes, BW 100 Kbit/sec, DLY 50000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 66.66.66.66 (Loopback666), destination 66.66.66.65
Tunnel Subblocks:
src-track:
Tunnel666 source tracking subblock associated with Loopback666
Set of tunnels with source Loopback666, 1 member (includes iterators), on interface <OK>
Tunnel protocol/transport GRE/IP
Key disabled, sequencing disabled
Checksumming of packets disabled
Tunnel TTL 255, Fast tunneling enabled
Tunnel transport MTU 1476 bytes
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Last input 00:07:00, output 00:02:39, output hang never
Last clearing of "show interface" counters 00:07:19
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
L2 Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes
L3 in Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes mcast
L3 out Switched: ucast: 0 pkt, 0 bytes mcast: 0 pkt, 0 bytes
20 packets input, 1944 bytes, 0 no buffer
  Received 0 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
26 packets output, 2504 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 unknown protocol drops
0 output buffer failures, 0 output buffers swapped out

**Troubleshooting Tips**

For troubleshooting information, contact Cisco Technical Assistance Center (TAC) at:
IPv6 over IPv4-GRE Tunnels

IPv6 traffic is carried over IPv4 generic routing encapsulation (GRE) tunnels using the standard GRE tunneling technique. As in the manually configured IPv6 tunnels, GRE tunnels are links between two points, with a separate tunnel for each link. The GRE tunnels provide stable connections that require regular secure communication between two edge routers or between an edge router and an end system. This feature supports VRF Aware IPv6 over IPv4-GRE Tunnel on the c7600.

Figure 13-6  Topology for VRF Aware IPv6 over IPv4-GRE

The VRF Aware IPv6 over IPv4 GRE tunnel must have ES+ line card towards the core facing side.

Restrictions for IPv6 over IPv4-GRE tunnel

Following restrictions apply to the IPv6 over IPv4-GRE tunnel:

- The IPv4 tunnel facing interface must be on the ES+ line card.
- The GRE tunnel key is not supported in the hardware.
The IPv4 fragmentation after tunnel encapsulation is not supported in the hardware.

The fragmented IPv4 packets for tunnel decapsulation is not supported in the hardware.

The IPv4 GRE keepalives are supported, but the IPv6 GRE keepalives are not supported.

The keepalives are not supported when the VRF instances configured using the `vrf forwarding` and `tunnel vrf` commands are different.

Due to EARL limitation, same source tunnels across VRF’s are not supported.

This feature is not SSO compliant.

With scaled configurations, when changing the tunnel mode from IPv6 over GRE to IPv6IP and on enabling the `mls mpls tunnel-recirc` command, the system displays an error message with a traceback.

## Configuring IPv6 over IPv4-GRE tunnel

The following sections describe how to configure IPv6 over IPv4-GRE tunnel on the c7600 platform:

- Configure IPv6 traffic over IPv4-GRE, page 13-79
- Configure VRF Aware IPv6 over IPv4-GRE Tunnel, page 13-82

### Configure IPv6 traffic over IPv4-GRE

Complete the following steps to configure IPv6 traffic over IPv4-GRE tunnel:

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface gigabitethernet slot/port**
5. **ipv6 address {ipv6-address/prefix-length \ prefix-name sub-bits/prefix-length}**
6. **exit**
7. **interface gigabitethernet slot/port**
8. **ip address ip-address**
9. **exit**
10. **interface loopback interface-number**
11. **ip address ip-address**
12. **exit**
13. **interface tunnel tunnel-number**
14. **ipv6 enable**
15. **ipv6 address {ipv6-address/prefix-length \ prefix-name sub-bits/prefix-length}**
16. **tunnel source {ip-address \ interface-type interface-number}**
17. **tunnel destination {hostname \ ip-address \ ipv6-address}**
18. **tunnel mode gre ip**
19. **exit**
### IPv6 over IPv4-GRE Tunnels

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv6 unicast-routing</td>
<td>Enables the forwarding of IPv6 unicast datagrams.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# ipv6 unicast-routing</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> interface gigabitethernet slot/port</td>
<td>Enters the interface configuration mode and specifies the Gigabit interface to configure.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# interface gigabitethernet 3/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> ipv6 address</td>
<td>Specifies the IPv6 address assigned to the interface, and enables IPv6 processing on the interface.</td>
</tr>
<tr>
<td>(ipv6-address</td>
<td>prefix-length</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-if)# ipv6 address 1::2/64</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# (config-if)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> interface gigabitethernet slot/port</td>
<td>Enters the interface configuration mode and specifies the Gigabit interface to configure.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# interface gigabitethernet 4/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> ip address ip-address</td>
<td>Assigns an IP address and subnet mask to the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-if)# ip address 10.1.1.1 255.255.255.0</td>
<td></td>
</tr>
</tbody>
</table>
### Step 9
**Command or Action**: exit

**Example**: Router(config-if)# exit

Exits interface configuration mode.

### Step 10
**Command or Action**: interface loopback

**Example**: Router(config)# interface Loopback 666

Enters interface configuration mode and names the new loopback interface.

**Note**: This command configures a loopback interface for the tunnel source.

### Step 11
**Command or Action**: ip address ip-address

**Example**: Router(config-if)# ip address 66.66.66.66 255.255.255.255

Assigns an IP address and subnet mask to the loopback interface.

### Step 12
**Command or Action**: exit

**Example**: Router(config-if)# exit

Exits interface configuration mode.

### Step 13
**Command or Action**: interface tunnel

**Example**: Router(config)# interface tunnel 666

Specifies a tunnel interface and enters the interface configuration mode.

**Note**: This command configures the IPv6 tunneling over IPv4 Transport.

### Step 14
**Command or Action**: ipv6 enable

**Example**: Router(config-if)# ipv6 enable

Enables IPv6 processing on an interface not configured with an explicit IPv6 address.

### Step 15
**Command or Action**: ipv6 address

**Example**: Router(config-if)# ipv6 address 3::1/120

Specifies the IPv6 address assigned to the interface, and enables IPv6 processing on the interface.

### Step 16
**Command or Action**: tunnel source

**Example**: Router(config-if)# tunnel source loopback 666

Specifies the source interface type and number for the tunnel interface.
Chapter 13  Configuring Layer 3 and Layer 4 Features

IPv6 over IPv4-GRE Tunnels

This example shows how to configure IPv6 traffic over IPv4-GRE tunnel:

```
Router# enable
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)# interface gigabitethernet 3/1
Router(config-if)# ipv6 address 1::2/64
Router(config-if)# exit
Router(config)# interface gigabitethernet 4/1
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# interface Loopback 666
Router(config-if)# ip address 66.66.66.66 255.255.255.255
Router(config-if)# exit
Router(config)# interface tunnel 666
Router(config-if)# ipv6 enable
Router(config-if)# ipv6 address 3::1/120
Router(config-if)# tunnel source loopback 666
Router(config-if)# tunnel destination 10.66.66.1
Router(config-if)# tunnel mode gre ip
Router(config-if)# end
```

Configure VRF Aware IPv6 over IPv4-GRE Tunnel

Complete the following steps to configure VRF Aware IPv6 over IPv4-GRE Tunnel:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>enable</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ipv6 unicast-routing</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>mls ipv6 vrf</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>vrf definition vrf name 1</td>
<td></td>
</tr>
</tbody>
</table>

---

**Command or Action**

**Purpose**

**Step 17**

tunnel destination
(host-name | ip-address | ipv6-address)

Specifies the destination address for a tunnel interface.

**Example:**

Router(config-if)# tunnel destination 10.66.66.1

**Step 18**

tunnel mode gre ip

Sets the encapsulation mode for the tunnel interface to GRE.

**Example:**

Router(config-if)# tunnel mode gre ip

**Step 19**

derm

Ends the current configuration session.

**Example:**

Router(config-if)# end

**Configuration Example**

Complete the following steps to configure VRF Aware IPv6 over IPv4-GRE Tunnel:

1. Enable the router.
2. Configure the terminal.
3. Enable IPv6 unicast routing.
4. Configure the Gigabit Ethernet interface 3/1 with IPv6 address 1::2/64.
5. Exit the interface configuration.
6. Configure the Gigabit Ethernet interface 4/1 with IP address 10.1.1.1/24.
7. Exit the interface configuration.
8. Configure the Loopback 666 interface with IPv6 address 66.66.66.66/64.
9. Exit the interface configuration.
10. Configure the tunnel interface 666.
11. Enable IPv6 on the tunnel.
12. Configure the tunnel source as loopback 666.
13. Configure the tunnel destination as 10.66.66.1.
14. Set the tunnel encapsulation mode to GRE.
15. Exit the configuration session.
Step 6  rd {ASN:nn | IP address: nn}
Step 7  route-target [import | export | both] {ASN:nn | IP address: nn}
Step 8  address-family ipv6
Step 9  exit
Step 10 address-family ipv4
Step 11 exit
Step 12 exit
Step 13 vrf definition vrf name 2
Step 14 rd {ASN:nn | IP address: nn}
Step 15 route-target [import | export | both] {ASN:nn | IP address: nn}
Step 16 address-family ipv4
Step 17 exit
Step 18 exit
Step 19 interface gigabitethernet slot/port
Step 20 vrf forwarding vrf name 1
Step 21 ipv6 address {ipv6-address/prefix-length | prefix-name sub-bits/prefix-length}
Step 22 exit
Step 23 interface gigabitethernet slot/port
Step 24 vrf forwarding vrf name 2
Step 25 ip address ip-address
Step 26 exit
Step 27 interface loopback interface-number
Step 28 vrf forwarding vrf name 2
Step 29 ip address ip-address
Step 30 exit
Step 31 interface tunnel tunnel-number
Step 32 vrf forwarding vrf name 1
Step 33 ipv6 address {ipv6-address/prefix-length | prefix-name sub-bits/prefix-length}
Step 34 tunnel source {ip-address | interface-type interface-number}
Step 35 tunnel destination {hostname | ip-address | ipv6-address}
Step 36 tunnel mode gre ip
Step 37 tunnel vrf vrf name 2
Step 38 end
### IPv6 over IPv4-GRE Tunnels

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
</tbody>
</table>
| enable            | Example:  
| Router# enable    | Enters global configuration mode. |
| **Step 3**        | Enables the forwarding of IPv6 unicast datagrams. |
| ipv6 unicast-routing | Example:  
| Router(config)# ipv6 unicast-routing | Configures a VRF instance and enters the VRF configuration mode. |
| **Step 4**        | Enables IPv6 globally in a VRF instance. |
| mls ipv6 vrf       | Example:  
| Router(config)# mls ipv6 vrf | Specifies an RD. |
| **Step 5**        | Specifies an RD. |
| vrf definition vrf name 1 | Example:  
| Router(config)# vrf definition VRF_RED | Creates a route-target extended community for a VRF instance. Route target extended community attributes are used to identify a set of sites and VRF instances that can receive routes with a configured route target. |
| **Step 6**        | Select an address family type for a VRF table and enters VRF address family configuration mode. This command configures the separate route-target policies for IPv6. |
| rd (ASN:nn | IP address: nn) | Example:  
| Router(config-vrf)# rd 1:1 | Selects an address family type for a VRF table and enters VRF address family configuration mode. This command configures the separate route-target policies for IPv6. |
| route-target [import | export | both](ASN:nn | IP address: nn) | Router(config-vrf)#route-target export 1:1  
| Router(config-vrf)#route-target import 1:1 | Specifies an RD. |
| **Step 7**        | Specifies an RD. |
| address-family ipv6 | Example:  
| Router(config-vrf)#address-family ipv6 | Specifies an RD. |
### IPv6 over IPv4-GRE Tunnels

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 9</td>
<td>exit</td>
<td>Exits the address family configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-vrf-af)#exit</td>
<td></td>
</tr>
<tr>
<td>Step 10</td>
<td>address-family ipv4</td>
<td>Selects an address family type for a VRF table and enters VRF address family configuration mode. This command configures the separate route-target policies for IPv4.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-vrf)#address-family ipv4</td>
<td></td>
</tr>
<tr>
<td>Step 11</td>
<td>exit</td>
<td>Exits the address family configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-vrf-af)#exit</td>
<td></td>
</tr>
<tr>
<td>Step 12</td>
<td>exit</td>
<td>Exits the VRF configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-vrf)#exit</td>
<td></td>
</tr>
<tr>
<td>Step 13</td>
<td>vrf definition vrf name 2</td>
<td>Configures a VRF instance and enters the VRF configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# vrf definition VRF_GREEN</td>
<td></td>
</tr>
<tr>
<td>Step 14</td>
<td>rd {ASN:nn</td>
<td>IP address: nn}</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-vrf)# rd 1:1</td>
<td></td>
</tr>
<tr>
<td>Step 15</td>
<td>route-target [import</td>
<td>export</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-vrf)#route-target import 1:1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-vrf)#route-target export 1:1</td>
<td></td>
</tr>
<tr>
<td>Step 16</td>
<td>address-family ipv4</td>
<td>Selects an address family type for a VRF table and enters VRF address family configuration mode. This command configures the separate route-target policies for IPv4.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-vrf)#address-family ipv4</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
---|---
**Step 17** | exit
**Example:**
Router (config-vrf-af)#exit
| Exits the address family configuration mode.

**Step 18** | exit
**Example:**
Router(config-vrf)#exit
| Exits the VRF configuration mode.

**Step 19** | interface gigabitethernet slot/port
**Example:**
Router(config)# interface gigabitethernet 3/1
| Enters the interface configuration mode and specifies the Gigabit interface to configure.
- *slot/port*—Specifies the location of the interface.

**Note**
This command configures the interface towards the IPv6 network.

**Step 20** | vrf forwarding vrf name 1
**Example:**
Router(config-if)#vrf forwarding VRF_RED
| Associates a VRF instance with an interface or a subinterface.

**Step 21** | ipv6 address
**Example:**
Router(config-if)# ipv6 address 1::2/64
| Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface.

**Step 22** | exit
**Example:**
Router# (config-if)# exit
| Exits interface configuration mode.

**Step 23** | interface gigabitethernet slot/port
**Example:**
Router(config)# interface gigabitethernet 4/1
| Enters the interface configuration mode and specifies the Gigabit interface to configure.
- *slot/port*—Specifies the location of the interface.

**Note**
This command configures the interface towards the IPv4 network.

**Step 24** | vrf forwarding vrf name 2
**Example:**
Router(config-if)#vrf forwarding VRF_GREEN
| Associates a VRF instance with an interface or a subinterface.
<table>
<thead>
<tr>
<th>Step 25</th>
<th>ip address ip-address</th>
<th>Assigns an IP address and subnet mask to the interface.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-if)#ip address 10.1.1.1 255.255.255.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 26</th>
<th>exit</th>
<th>Exits interface configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-if)# exit</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 27</th>
<th>interface loopback interface-number</th>
<th>Enters interface configuration mode and names the new loopback interface.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note:</td>
<td>This command configures a loopback interface for the tunnel source</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# interface Loopback 666</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 28</th>
<th>vrf forwarding vrf name 2</th>
<th>Associates a VRF instance with an interface or a subinterface.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-if)#vrf forwarding VRF_GREEN</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 29</th>
<th>ip address ip-address</th>
<th>Assigns an IP address and subnet mask to the loopback interface.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-if)#ip address 66.66.66.66 255.255.255.255</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 30</th>
<th>exit</th>
<th>Exits interface configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-if)# exit</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 31</th>
<th>interface tunnel tunnel-number</th>
<th>Specifies a tunnel interface and enters the interface configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note:</td>
<td>This command configures the IPv6 tunneling over IPv4 Transport.</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# interface tunnel 666</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 32</th>
<th>vrf forwarding vrf name 1</th>
<th>Associates a VRF instance with an interface or a subinterface.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note:</td>
<td>This command specifies the VRF instance to which the tunnel belongs, that is, the VRF instance used for IPv6 overlay address lookup.</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-if)#vrf forwarding VRF_RED</td>
<td></td>
</tr>
</tbody>
</table>
### IPv6 over IPv4-GRE Tunnels

**Chapter 13** Configuring Layer 3 and Layer 4 Features

**IPv6 over IPv4-GRE Tunnels**

**Step 33**

`ipv6 address
(ipv6-address|prefix-length | prefix-name sub-bits | prefix-length)`

**Example:**

Router(config-if)# ipv6 address 3::1/120

**Purpose:** Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface.

**Step 34**

`tunnel source (ip-address | interface-type interface-number)`

**Example:**

Router(config-if)# tunnel source loopback 666

**Purpose:** Specifies the source interface type and number for the tunnel interface.

**Step 35**

`tunnel destination (host-name | ip-address | ipv6-address)`

**Example:**

Router(config-if)# tunnel destination 10.66.66.1

**Purpose:** Specifies the destination address for a tunnel interface.

**Step 36**

`tunnel mode gre ip`

**Example:**

Router(config-if)# tunnel mode gre ip

**Purpose:** Sets the encapsulation mode for the tunnel interface to GRE.

**Step 37**

`tunnel vrf vrf name 2`

**Example:**

Router(config-if)# tunnel vrf VRF_GREEN

**Purpose:** Configures a VRF instance with a specific tunnel destination, interface or a subinterface.

**Note**

This command specifies the VRF instance used for tunnel IPv4 transport address lookup, that is, the tunnel source and the tunnel destination.

**Step 38**

`end`

**Example:**

Router(config-if)# end

**Purpose:** Ends the current configuration session.

### Configuration Example

This example shows how to configure VRF Aware IPv6 over IPv4-GRE Tunnel:

```bash
Router# enable
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)# mls ipv6 vrf
Router(config)# vrf definition VRF_RED
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv6
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv4
```
IPv6 over IPv4-GRE Tunnels

```
Router(config-vrf-af)# exit
Router(config-vrf)# exit
Router(config)# vrf definition VRF_GREEN
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# exit
Router(config-vrf)# exit
Router(config)# interface gigabitethernet 3/1
Router(config-if)# vrf forwarding VRF_RED
Router(config-if)# ipv6 address 1::2/64
Router(config-if)# exit
Router(config-if)# interface gigabitethernet 4/1
Router(config-if)# vrf forwarding VRF_GREEN
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# exit
Router(config-if)# interface Loopback 666
Router(config-if)# vrf forwarding VRF_GREEN
Router(config-if)# ip address 66.66.66.66 255.255.255.255
Router(config-if)# exit
Router(config-if)# interface tunnel 666
Router(config-if)# vrf forwarding VRF_RED
Router(config-if)# ipv6 address 3::1/120
Router(config-if)# tunnel source loopback 666
Router(config-if)# tunnel destination 10.66.66.1
Router(config-if)# tunnel mode gre ip
Router(config-if)# tunnel vrf VRF_GREEN
Router(config-if)# end
```

Verifying the Configuration

Use these commands to verify the configuration of IPv6 over IPv4-GRE tunnel on the c7600:

```
Router# show platform npc ipv6ogre interface tunnel 666
Tunnel666 is up, line protocol is up
   Hardware is Tunnel
       MTU 0 bytes, BW 10000000 Kbit/sec, DLY 0 usec,
       reliability 255/255, txload 1/255, rxload 1/255
   Encapsulation TUNNEL, loopback not set
   Keepalive set (10 sec)
   Tunnel source 66.66.66.66 (Loopback666), destination 66.66.66.65
   Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
   Checksumming of packets disabled, vip tunneling enabled
   Last input never, output never, output hang never
   Last clearing of "show interface" counters 00:08:54
   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 no buffer
       Received 0 broadcasts (0 IP multicasts)
       0 runts, 0 giants, 0 throttles
       0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
       0 packets output, 0 bytes, 0 underruns
       0 output errors, 0 collisions, 0 interface resets
       0 unknown protocol drops
       0 output buffer failures, 0 output buffers swapped out
   Platform information
       Tunnel vlan : 1026
       Tun rsvd vlan : 1025
       Phy vlan : 1017
       Tunnel id : 0
       HPLA addr : 0x23AF10A0
```
IPv6 over IPv4-GRE Tunnels

Router# show platform npc ipv6ogre egress-table 1026
IPv6OGRE egress table entry
eg_entry->match_cond = 1
eg_entry->ent_valid = 1
eg_entry->phy_vlan = 1017
eg_entry->src_ip = 66.66.66.66
eg_entry->dst_ip = 66.66.66.65
eg_entry->smac = 0012.44dc.9000
eg_entry->dmac = 0018.7468.0000
eg_entry->eg_stats_id = 639626 0x0009C28A
Raw dump
value: 00 00 3F 93 68 74 18 00 12 00 00 00 00 90 dc 44 ..?.ht.........\D
value: aa 45 00 08 42 42 42 42 42 42 42 42 42 42 00 09 C2 8A *E..BBBBBABB..B.

Router# show platform npc ipv6ogre tcam 1026
Dumping tcam for 1026 on NP 0
Key Decode :
Source      IP : 66.66.66.65 Mask : 00000000
Destination IP : 66.66.66.66 Mask : 00000000
Feature id     : 3 Mask : 00
Result Decode :
Vlan            : 1025
Statistics ID   : 0x9C287

Raw output :
g_vmr.value : 42 42 42 42 42 42 42 42 42 03 00 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF FF 64 A0
g_vmr.mask  : 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF FF 64 A0
g_vmr.result: 04 01 01 03 00 09 C2 87

Dumping tcam for 1026 on NP 1
Key Decode :
Source      IP : 66.66.66.65 Mask : 00000000
Destination IP : 66.66.66.66 Mask : 00000000
Feature id     : 3 Mask : 00
Result Decode :
Vlan            : 1025
Statistics ID   : 0x9C28B

Raw output :
g_vmr.value : 42 42 42 42 42 42 42 42 42 03 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF FF 64 A0
g_vmr.mask  : 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF FF 64 A0
g_vmr.result: 04 01 01 03 00 09 C2 8B

Router# show platform npc ipv6ogre xlif 1026
Egress XLIF table fields

Feature common enable: 0x1
Feature enable: 0x1
Feature bits: 0x04
Control common bits: 0x00
Control feature bits: 0x00
Control rewrite opcode: 0x00
Port: 0x4
Match cond 0x1
Entry valid: 0x1
Optimal Path en : 0x0
Dbus VLAN: 1017
QoS policy ID: 0
ACL ID: 0
Statistics ID: 0
Inner rewrite VLAN: 0
Outer rewrite VLAN: 0
QoS flow ID: 0
IP Session en : 0
Chapter 13 Configuring Layer 3 and Layer 4 Features

IPv6 over IPv4-GRE Tunnels

Feature data 0
Intf etype: 0x40C40010
Multicast enable: 0x00000001
Post Filter Opcode 0x00000004
Pre Filter Opcode 0x00000000
Pre Tag Outer 0x00000010
Pre Tag Inner 0x000000C4
Post Filter Vlan high 0x00000414
Post Filter Vlan low 0x00000242
Post Filter Vlan outer 0x00000242
EVC - MST: 0x0
EVC etype 0x03F9
CFM MEP Level 0x00000004
CFM MIP Level 0x00000002
CFM disable 0x0
MIP filtering 0x01
block data: 0x0
sacl: 0x0
sacl index: 0x0000
sacl statid: 0x0000100
Span Enable: 0x0

Egress XLIF table fields

Feature common enable: 0x1
Feature enable: 0x1
Feature bits: 0x0001
Control common bits: 0x0000
Control feature bits: 0x0000
Control rewrite opcode: 0x0000
Port: 0x04
Match cond: 0x01
Entry valid: 0x01
Optimal Path en: 0x01
Dbus VLAN: 1017
QoS policy ID: 0
ACL ID: 0
Statistics ID: 0
Inner rewrite VLAN: 0
Outer rewrite VLAN: 0
QoS flow ID: 0
IP Session en: 0
Feature data 0 0x000C40010
Intf etype 0x00000000
Multicast enable: 0x00000000
Post Filter Opcode 0x00000000
Pre Filter Opcode 0x00000000
Pre Tag Outer 0x00000000
Pre Tag Inner 0x00000000
Post Filter Vlan high 0x00000000
Post Filter Vlan low 0x00000000
Post Filter Vlan outer 0x00000000
EVC - MST: 0x0
EVC etype 0x0000
CFM MEP Level 0x00000000
CFM MIP Level 0x00000000
CFM disable 0x0
MIP filtering 0x0
block data: 0x0
block_l2bpdu: 0x0
sacl: 0x0
sacl index: 0x0000
sacl statid: 0x00000
IPv6 Policy Based Routing

IPv6 policy-based routing (PBR) provides a flexible mechanism to route packets and define policy for the traffic flows. It extends and complements the existing mechanisms provided by routing protocols. PBR also provides a basic packet-marking capability.

PBR performs the following tasks:
- Classifies traffic based on extended access list criteria. It provides access to lists and then establishes the match criteria.
- Sets IPv6 precedence bits and enables the network to differentiate classes of service.
- Routes packets to specific traffic-engineered paths. You can route the packets to allow a specific quality of service (QoS) through the network.

The Cisco 7600 Series Router implements this feature using the Earl7 forwarding engines capability to classify traffic through an Access Control List (ACL) Ternary Content Addressable Memory (TCAM) lookup. The ACL TCAM lookup classifies traffic based on the combination of a variety of Layer 3 and Layer 4 traffic parameters. Once classified, the ACL TCAM drives results for matching flows. The Feature Manager (FM) component converts the route map policy configured on an interface into a series of values, masks and results (VMRs) and programs these in the ACL TCAM.

Policy Based Routing

All packets received on a PBR-enabled interface are passed through enhanced packet filters known as route maps. Route maps are composed of statements that are marked as permit or deny, and they are interpreted in these ways:
- If a packet matches all match statements for a route map that is marked as permit, the router subjects the packet to PBR using the set statements.
- If the packet matches any match statements for a route map that is marked as deny, the router does not subject the packet to PBR and forwards it normally.
- If the statement is marked as permit and the packets do not match any route map statements, the router sends the packets back through the normal forwarding channels and performs destination-based routing.

Packet Matching

The IPv6 PBR match criterion for a sequence is specified through a combination of IPv6 access-lists and packet length operations. Match statements are evaluated first by the criteria specified in the match ipv6 address command and then by criteria specified in the match length command. Therefore, if both an
ACL and a length statement are used, a packet is first subjected to an ACL match. Only packets that pass the ACL match are subjected to the length match. Finally, only packets that pass both the ACL and the length statement are policy routed.

**Packet Forwarding Using Set Statements**

PBR for IPv6 packet forwarding is controlled using a number of set statements in the PBR route map. Listed below are the forwarding actions in order of decreasing priority, and the manner in which these options are reflected in the result from the VMRs programmed in the ACL TCAM. When more than one kind of packet forwarding action is specified in a sequence, the one with the highest priority is chosen.

<table>
<thead>
<tr>
<th>Set Statement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>set vrf vrf name</code></td>
<td>Specifies the VPN Routing and Forwarding (VRF) instance to which the packet should be sent, based on packet attributes. By default the VRF that a packet is forwarded on is the same as the VRF that receives the packet.</td>
</tr>
<tr>
<td><code>set ipv6 next-hop next-hop ipv6 address</code></td>
<td>Specifies the next hop for the packet. The next hop must be present in the Routing Information Base (RIB); it must be directly connected, and it must be a global IPv6 address. If the next hop is invalid, the set statement is ignored.</td>
</tr>
<tr>
<td><code>set interface next-hop interface</code></td>
<td>Specifies the next hop interface for the packet. A packet is forwarded out of a specified interface. An entry for the packet destination address must exist in the IPv6 RIB, and the specified output interface must be in the path set. If the interface is invalid, the set statement is ignored.</td>
</tr>
<tr>
<td><code>set ipv6 default next-hop default next-hop ipv6 address</code></td>
<td>Specifies the connected next hop for the packet if the usual forwarding method fails to produce the default result. It must be a global IPv6 address. This set statement is used only when there is no explicit entry for the packet destination in the IPv6 RIB.</td>
</tr>
<tr>
<td><code>set default interface default next-hop interface</code></td>
<td>Specifies the default next-hop interface, from which the matching packets are forwarded if the usual forwarding method fails to produce a result. This set statement is used only when there is no explicit entry for the packet destination in the IPv6 RIB.</td>
</tr>
</tbody>
</table>

**Restrictions for IPv6 PBR**

Following restrictions apply to the IPv6 PBR:

- Match length is not supported in the hardware, and the PBR is applied to the software.
- Packet marking actions are not supported in the hardware, and packets requiring marking due to PBR are punted to the software.
- Set interface is supported in the hardware only for the serial interface. Other interfaces are supported on the software.
- Packets containing an IPv6 hop-by-hop header need to be examined by the router and are punted to the software. Such packets are subjected to PBR in the software.
IPv6 Policy Based Routing

- PBR policies using access-lists matching on IPv6 flow label, DSCP value and extension headers such as, routing, mobility, destination headers cannot be fully classified in the hardware, and are punted to the software after partial classification.

- It is not possible to completely classify traffic in hardware, when access-lists matching on non compressible addresses are used. In such cases, the PBR is applied to the software.

- On Tycho based systems, fragment packets that require matching on layer 4 protocol are punted to the software.

- IPv6 PBR on SVI interfaces is applied to the software, and hardware provides only partial classification.

- IPv6 PBR when applied to hardware will also be applied on packets destined to a router address.

- A set next-hop action where the next-hop is at the other end of a tunnel is not supported in the hardware.

- For set interface and set default interface, the interface should be a point-to-point one.

- PBR is not applied to multicast traffic and the traffic destined to link local addresses.

- When there is no traffic flow, the TCAM entry does not change from punt to policy-route.

**Configuring IPv6 PBR**

To configure, verify and troubleshoot the IPv6 PBR, see: Configuring IPv6 PBR.