ERICSSON

Configuring Forward Policies

SYSTEM ADMINISTRATOR GUIDE

27/1543-CRA 119 1170/1-V1 Uen G

Contents

1 Overview

This document provides an overview of forwarding policy features supported by the SmartEdge router and describes the tasks used to configure, monitor, and administer the forward policies. This document also provides configuration examples of forward policies.

This document applies to both the Ericsson SmartEdge® and SM family routers. However, the software that applies to the SM family of systems is a subset of the SmartEdge OS; some of the functionality described in this document may not apply to SM family routers.

For information specific to the SM family chassis, including line cards, refer to the SM family chassis documentation.

For specific information about the differences between the SmartEdge and SM family routers, refer to the Technical Product Description *SM Family of Systems* (part number 5/221 02-CRA 119 1170/1) in the **`Product Overview`** folder of this Customer Product Information library.

A forward policy applies only to IP traffic. A forward policy can be a combination of three actions:

- Mirroring

  Mirroring duplicates packets to an output circuit different from a normal forwarding circuit. Mirrored traffic (packets forwarded, dropped, or both) is typically sent to a packet sniffer (or similar device) so that traffic patterns can be analyzed. You can mirror all traffic, a sampling of traffic, or mirror only IP packet headers (depending on the data type). You can mirror both incoming and outgoing packets.

  Note: Forward policy does not support mirror action for IPv6 traffic.

- Redirect

  Redirect forwards packets to IP addresses that are different than their original destination. You can redirect incoming packets only.

  Note: Layer 2 circuits do not support redirect.

  Note: Forward policies currently only support local (HTTP) or next-hop redirect modes for IPv6 traffic.

Note: IPv6 nodes on the same link use Neighbor Discovery (ND) protocol to discover each other's presence. To allow ND notifications, incoming ND messages Router Solicitation (133), Neighbor Solicitation (135) and Redirect (137) should be excluded from redirects. However, ICMPv6 Echo Request (128) message should be redirected for a proper 'ping' operation. Redirect approaches for ICMPv6 messages are:

- Select only non-ICMPv6 traffic for redirect action.
- Use a distinct rule set for specific ICMPv6 messages in the IPv6 Policy ACL that matches a corresponding "exclude" (no action) IPv6 forwarding class.

- Drop

  The drop function specifies that particular packets are dropped, rather than forwarded; you can drop incoming packets only.

You can apply forward policies at one of two levels or at both levels simultaneously. One level applies to all packets on a circuit and is referred to as circuit-based forwarding. Another level applies only to a specific class of packets traveling across a circuit and is referred to as class-based forwarding.

These levels of forward policies are described in the following sections:

1.1 Circuit-Based Forwarding

When you attach a forward policy that does not include a policy access control list (ACL) to a circuit, all traffic traveling over the circuit is treated in one manner; that is, it is mirrored, redirected, or dropped.

1.2 Class-Based Forwarding

A policy ACL classifies packets using classification statements (rules). Each policy ACL supports up to eight unique classes. You can classify a packet according to its:

- IP precedence value
- Protocol number
- IP source and destination address
- Internet Control Message Protocol (ICMP) attributes
- Internet Group Management Protocol (IGMP) attributes
- Transmission Control Protocol (TCP) attributes
- User Datagram Protocol (UDP) attributes

Note: Class-based forwarding is only supported on Layer 3 circuits. Class forwarding is not supported for IPV6 LAC and VPLS, including bridging.

To configure class-based forwarding for a circuit, you apply a policy ACL to a forward policy, specify the action that you want the policy to take for each class, and then attach the forward policy to the circuit. For more information about policy ACLs, see *Configuring ACLs*.

Note: If you do not specify an action for a class that is defined in the policy ACL, the SmartEdge router considers the class to be the default class.

1.3 Circuit- and Class-Based Forwarding

You can combine circuit-based and class-based forwarding, so that a class of packets can be treated in one manner, dependent on a policy ACL, while all remaining packets traveling across the circuit are treated strictly according to the forward policy conditions.

1.4 Forward Policy Support Per Circuit Type

Forward policy support varies per circuit type. For example, Layer 3 circuits have a principal binding or application for forwarding IP packets, and are configured using the **`bind interface`**, **`bind subscriber`**, and **`bind auth`** commands. Layer 2 circuits have a principal binding or application for forwarding encapsulated frames which may be IP or other Layer 3 protocols. Examples of Layer 2 circuits are L2VPNs (Layer 2 VPNs, Layer 2 XC (cross-connected) circuits, VLAN bridged circuits, and L2TP (Layer 2 Tunneling Protocol) LAC (L2TP access concentrator) session circuits.

The following Layer 3 circuits and traffic support forwarding policies: all circuits that carry IP routed traffic. Operating system Release 6.1.4.2 and higher releases also allow the following Layer 2 circuits and traffic to support forwarding policies: attachment circuits of all L2VPN circuits and XC circuits.

Note: The following Layer 2 circuits and traffic do not currently support forwarding policies: all bridged or VPLS circuits and LAC session circuits.

See *mirror destination* for the forward policy functionality available for Layer 2 and Layer 3 circuits.

1.5 IPv6 Forwarding Services

IPv6 forwarding services support the following functions:

- Filter-ID and Ascend-Data-Filter IPv6 ACL attributes for subscribers
- Dynamic policy filter (DPF) and dynamic QoS parameter (DQP) IPv6 policy ACL attributes when used with RADIUS Guided (RG) policies

- RADIUS Service Engine (RSE) support for IPv6 ACL attributes and service accounting
- RSE session rate limiting on LACs for both IPv4 and IPv6 traffic
- IPv6 and dual stack subscribers
- IPv6 ACL per rule counting

A forward policy defines actions on IPv6 traffic that match classifications rules. These actions can be applied to the traffic in a circuit and/or traffic that matches an IPv6 ACL. The forward class actions are: drop, mirror and redirect. Supported traffic types are:

- Point-to-Point Protocol over Ethernet (PPPoE) and PPPoE over access link aggregation group (LAG)
- L2TP network server LNS
- Static circuits, including dot1q, QinQ SVLAN and QinQ CVLAN
- Circuit creation on demand (CCoD)
- Nonsubscriber circuits carrying IPv6 traffic

Note: IPv6 forwarding does not apply to LAC and VPLS, including bridging.

2 Configuration and Operations Tasks

Note: In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, find and select the command in the *Command List*.

2.1 Configuring a Forward Policy to Handle IPv4 Traffic

To configure an IPv4 forward policy for circuit-based forwarding, for class-based forwarding, or for circuit- and class-based forwarding, perform the following tasks; enter all commands in forward policy configuration mode, unless otherwise noted:

1. Create or select a policy and access forward policy configuration mode using the *forward policy* command in global configuration mode.
2. Redirect incoming packets not associated with a class with the *redirect destination local (HTTP)* command (to the specified output destination) or the *redirect destination ip next-hop* command (to a next-hop IP address)
3. Drop incoming packets not associated with a class with the *drop (forward policy)* command.
4. Mirror specified incoming or outgoing packets not associated with a class to a specified output destination with the *mirror destination* command.
5. Optional. Configure class-based forwarding for this policy; see Section 2.2 on page 6.
6. Specify the destination circuit using the *forward output (Circuit)* command.

   Enter this command in ATM PVC, Frame Relay PVC, GRE tunnel, or port configuration mode.

   Select a different circuit from the circuits you have configured for the traffic being mirrored or redirected.
7. Attach the policy to a circuit with one of the following commands in ATM OC, ATM PVC, dot1q PVC, DS-0 group, Frame Relay PVC, port, or subscriber configuration mode:

   - To incoming traffic—*forward policy in*

     Only incoming packets can be redirected or dropped. Both incoming and outgoing packets can be mirrored.

   - To outgoing traffic—*forward policy out*

     Note: A redirect may be performed on ingress packets only.

2.2 Applying an IPv4 Policy ACL to a Forward Policy

To apply an IPv4 policy ACL to a forward policy for class-based forwarding, perform the following tasks; enter all commands in policy group class configuration mode, unless otherwise noted:

Note: Policy ACLs are only supported on Layer 3 circuits.

1. Apply a policy ACL to the forward policy, and access policy group configuration mode using the *ip access-group* command in forward policy configuration mode.
2. Specify a class and access policy group class configuration mode with the *class* command in policy group configuration mode.

   For class-based forwarding to occur, the class name must match one of the class names defined in the policy ACL.
3. Optional. Redirect incoming packets associated with the class with one of the following commands:
   - To the specified output destination—*redirect destination circuit*
   - To a next-hop IP address—*redirect destination ip next-hop*
4. Optional. Drop incoming packets associated with the class using the *drop (forward policy)* command.
5. Mirror specified packets associated with the class to a specified output destination using the *mirror destination* command.

Note: The **`redirect destination local`** command is used only for HTTP redirect and is described in *Configuring HTTP Redirect*.

2.3 Configuring a Forward Policy to Handle IPv6 Traffic

To configure a forward policy for circuit-based forwarding, for class-based forwarding, or for circuit- and class-based forwarding for IPv6 or dual-stack traffic, perform the following tasks; enter all commands in forward policy configuration mode, unless otherwise noted.

1. Create or select a policy and access forward policy configuration mode using the *forward policy* command in global configuration mode.
2. Redirect incoming packets not associated with a class with one of the following tasks:
   - To the specified output destination (local mode only) using the *redirect destination local* command.
   - To the specified IPv6 next-hop address using the *redirect destination ipv6 next-hop* command.

3. Drop incoming packets not associated with a class using the *drop (forward policy)* command.

4. Optional. Configure class-based forwarding for this policy; see Section 2.2 on page 6.

5. Attach the policy in subscriber configuration mode using the *forward policy in* command. You can attach a policy to inbound traffic only.

   You can enable policy ACL rule counters for subscribers by including the **`acl-counters`** keyword. ACL counters can impact router performance; use caution when enabling them.

Note: Forward policies currently only support the following actions for IPv6 packets: local (HTTP) or next-hop redirect and drop. Because HTTP redirect actions are only supported for subscriber circuits, there is no benefit to applying a forward policy to a non-subscriber circuit which carries only IPv6 traffic.

2.4 Applying an IPv6 Policy ACL to a Forward Policy for Class-Based Forwarding

To apply an IPv6 policy ACL to a forward policy for class-based forwarding, perform the following tasks. Enter all commands in policy group class configuration mode, unless otherwise noted.

1. Apply an IPv6 policy ACL to the forward policy, and access policy group configuration mode using the *ipv6 access-group* command in forward policy configuration mode.

2. Specify a class and access policy group class configuration mode using the *class* command in policy group configuration mode.

   For class-based forwarding to occur, the class name must match one of the class names defined in the policy ACL.

3. Optional. In local mode, redirect incoming packets associated with the class, using the *redirect destination local* command.

4. Optional. Redirect incoming packets associated with the class to an IPv6 next-hop address using the *redirect destination ipv6 next-hop* command.

5. Optional. Drop incoming packets associated with the class using the *drop (forward policy)* command.

2.5 Operations Tasks

To monitor, troubleshoot, and administer forward policies, perform the tasks described in Table 1. Enter the **`clear`** command in exec mode; enter the **`show`** commands in any mode.

For information about enabling ACL counters, see *Configuring ACLs*.

Table 1 Forward Policy Operations Tasks

Task	**Root Command**
Clears information about ACLs used with forward policies that are attached to ports, channels, or circuits.	*clear ip access-group*
Display information about ACLs used with forward policies that are attached to ports, channels, or circuits.	*show ip access-group* *show access-group forward* *show access-group ipv6 forward*
Display the configuration of forward policies.	*show configuration forward*
Display information about configured forward policies.	*show forward policy*

3 Configuration Examples

This section provides forward policy configuration examples.

3.1 Traffic Mirroring

The following example implements traffic mirroring for:

- Web traffic-to-POS port 13/1
- Forwarded UDP traffic-to-POS port 13/2
- Dropped IP packets-to-Ethernet port 4/1 not more frequently than once every three seconds
- Other traffic-to-POS port 13/3

Traffic comes in through the interface, **incoming_traffic**, and leaves the router through the interface, **normal_traffic**.

Figure 1 displays the network topology for this example.

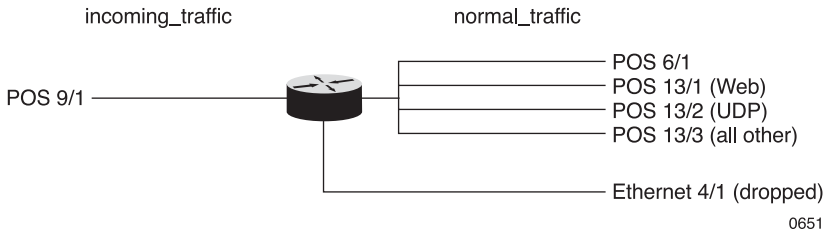

Figure 1 Basic Traffic Mirroring Network Topology (651)

The following example shows how to configure the interface.

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#interface e1
[local]Redback(config-if)#ip address 31.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface incoming_traffic
[local]Redback(config-if)#ip address 51.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface normal_traffic
[local]Redback(config-if)#ip address 41.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface p1
[local]Redback(config-if)#ip address 21.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface p2
[local]Redback(config-if)#ip address 22.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface p3
[local]Redback(config-if)#ip address 23.1.1.1/24
```

The following example shows how to configure the policy ACL.

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#policy access-list PBR_ACL
[local]Redback(config-access-list)#seq 10 permit tcp any eq www any class WEB
[local]Redback(config-access-list)#seq 20 permit tcp any any eq www class WEB
[local]Redback(config-access-list)#seq 30 permit udp any class UDP
[local]Redback(config-access-list)#seq 40 permit ip any class IP
```

The following example shows how to configure the forward policy.

```
[local]Redback#config
[local]Redback(config)#forward policy MirrorPolicy
[local]Redback(config-policy-frwd)#mirror destination DroppedTraffic dropped sampling 3000
[local]Redback(config-policy-frwd)#ip access-group PBR_ACL local
[local]Redback(config-policy-group)#class WEB
[local]Redback(config-policy-group-class)#mirror destination WebTraffic all
[local]Redback(config-policy-group-class)#exit
[local]Redback(config-policy-group)#class UDP
[local]Redback(config-policy-group-class)#mirror destination UdpTraffic forwarded
[local]Redback(config-policy-group-class)#exit
[local]Redback(config-policy-group)#class IP
[local]Redback(config-policy-group-class)#mirror destination IpTraffic all
```

The following configuration shows how to attach the forward policy to incoming circuits and define the forward output destinations:

```
[local]Redback#config
[local]Redback(config)#port ethernet 4/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface e1 local
[local]Redback(config-port)#forward output DroppedTraffic
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 6/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface normal_traffic local
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 9/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface incoming_traffic local
[local]Redback(config-port)#forward policy MirrorPolicy in
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 13/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface p1 local
[local]Redback(config-port)#forward output WebTraffic
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 13/2
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface p2 local
[local]Redback(config-port)#forward output UdpTraffic
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 13/3
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface p3 local
[local]Redback(config-port)#forward output IpTraffic
```

3.2 Layer 2 Mirroring for Attachment Circuits

The following examples show how to implement traffic mirroring for:

- Mirroring XC (cross-connect) traffic onto a local circuit
- Mirroring L2VPN (Layer 2 VPN) cross-connect AC (Attachment Circuit) traffic onto a GRE (Generic Routing Encapsulation) tunnel

The following example shows how to configure mirroring XC traffic onto a local circuit (applied on the port).

```
[local]Redback#config
[local]Redback(config)#forward policy xc-policy
[local]Redback(config-policy-frwd)#mirror destination local-ckt1 all l2-frames
[local]Redback(config-policy-frwd)#port ethernet 2/1
[local]Redback(config-port)#encap dot1q
[local]Redback(config-port)#dot1q pvc 1
[local]Redback(config-dot1q-pvc)#bind bypass
[local]Redback(config-dot1q-pvc)#forward policy xc-policy in
[local]Redback(config-dotq1-pvc)#port ethernet 3/1
[local]Redback(config-port)#encap dot1q
[local]Redback(config-port)#dot1q pvc 100 encap 1qtunnel
[local]Redback(config-dot1q-pvc)#dot1q pvc 100:1
[local]Redback(config-dot1q-pvc)#bind bypass
[local]Redback(config-dot1q-pvc)#port ethernet 10/2
[local]Redback(config-port)#forward output local-ckt1
[local]Redback(config-port)#xc 2/1 vlan 1 to 3/1 vlan 100:1
[local]Redback(config)#end
```

In this configuration, traffic comes in through interface `2/1 vlan 1` interface, is forwarded to the `3/1 vlan 100:1` interface, and is also mirrored to the `10/2` port.

The following example shows how to configure mirroring L2VPN cross-connect AC traffic onto a GRE tunnel:

```
[local]Redback#config
[local]Redback(config)#forward policy cross-connect-policy
[local]Redback(config-policy-frwd)#mirror destination gre-tunnel1 all ip-datagrams
[local]Redback(config-policy-frwd)#end
[local]Redback(config)#
[local]Redback(config)#tunnel gre tunnel01
[local]Redback(config-tunnel)#peer-end-point local 1.1.1.10 remote 1.1.1.5
[local]Redback(config-tunnel)#bind interface if2 local
[local]Redback(config-tunnel)#forward output gre-tunnel1
[local]Redback(config-tunnel)#context local
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#xc-group 1
[local]Redback(config-l2vpn-xc-group)#xc 2/1 vlan 1 vc-id 10 peer 2.2.2.2
[local]Redback(config-l2vpn-xc-group)#port ethernet 2/1
[local]Redback(config-port)#dot1q pvc 1
[local]Redback(config-dot1q-pvc)#l2vpn local
[local]Redback(config-dot1q-pvc)#forward policy cross-connect-policy out
[local]Redback(config-dot1q-pvc)#forward policy cross-connect-policy in
```

In this configuration, the traffic that comes in through **2/1 vlan 1** port is forwarded to the **vc-id 10** circuit and is mirrored to the **tunnel01** GRE tunnel. The traffic that comes in through the **vc-id 10** circuit is forwarded to the **2/1 vlan 1** port and is also mirrored to the **tunnel01** GRE tunnel.

3.3 Traffic Redirect

The following example shows how to implement traffic redirection for:

- Web traffic-to-network 100.1.1.0 with load balancing
- Forwarded UDP traffic-to-network 100.1.1.0 with load balancing
- Other TCP traffic-to-POS port 13/3 (multipath redirect)
- Protocol Independent Multicast (PIM) traffic-to-Ethernet port 4/1 (redirect to circuit)

This configuration allows all other traffic flow in the normal path. Traffic comes in through the interface, **incoming_traffic**, and leaves the router through the interface, **normal_traffic**. Figure 2 displays the network topology for this example.

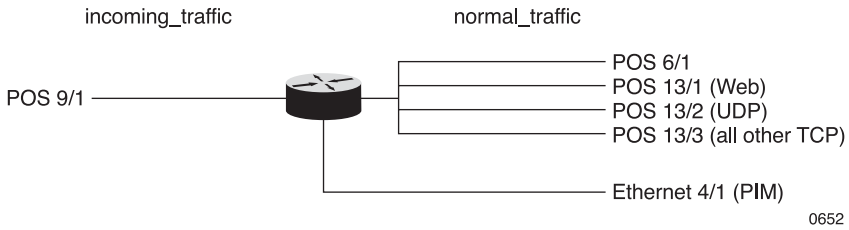

Figure 2 Basic Traffic Redirect Network Topology (652)

Note: Traffic redirect is only supported on Layer 3 circuits.

The following example shows how to configure the interface.

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#interface e1
[local]Redback(config-if)#ip address 31.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface incoming_traffic
[local]Redback(config-if)#ip address 51.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface normal_traffic
[local]Redback(config-if)#ip address 41.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface p1
[local]Redback(config-if)#ip address 21.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface p2
[local]Redback(config-if)#ip address 22.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface p3
[local]Redback(config-if)#ip address 23.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#ip route 100.1.1.0/24 21.1.1.2
[local]Redback(config-ctx)#ip route 100.1.1.0/24 22.1.1.2
```

The following example shows how to configure the policy ACL.

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#policy access-list PBR_Redirect_ACL
[local]Redback(config-access-list)#seq 10 permit tcp any eq www any class WEB
[local]Redback(config-access-list)#seq 20 permit tcp any any eq www class WEB
[local]Redback(config-access-list)#seq 30 permit tcp any class TCP
[local]Redback(config-access-list)#seq 40 permit udp any class UDP
[local]Redback(config-access-list)#seq 50 permit pim any class PIM
```

The following example shows how to configure the forward policy.

```
[local]Redback(config)#forward policy RedirectPolicy
[local]Redback(config-policy-frwd)#ip access-group PBR_Redirect_ACL local
[local]Redback(config-policy-group)#class WEB
[local]Redback(config-policy-group-class)#redirect destination ip next-hop 100.1.1.0
[local]Redback(config-policy-group-class)#exit
[local]Redback(config-policy-group)#class UDP
[local]Redback(config-policy-group-class)#redirect destination ip next-hop 100.1.1.0
[local]Redback(config-policy-group-class)#exit
[local]Redback(config-policy-group)#class PIM
[local]Redback(config-policy-group-class)#redirect destination circuit PIM_OUT
[local]Redback(config-policy-group-class)#exit
[local]Redback(config-policy-group)#class TCP
[local]Redback(config-policy-group-class)#redirect destination ip next-hop
23.1.1.11 23.1.1.12 23.1.1.13 23.1.1.14
```

The following configuration shows how to attach the forward policy to an incoming circuit and define the forward output destinations:

```
[local]Redback(config)#port ethernet 4/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface e1 local
[local]Redback(config-port)#forward output PIM_OUT
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 6/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface normal_traffic local
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 9/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface incoming_traffic local
[local]Redback(config-port)#forward policy RedirectPolicy in
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 13/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface p1 local
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 13/2
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface p2 local
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 13/3
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface p3 local
```

3.3.1 Example IPv4 and IPv6 Redirection for Default Forwarding Class

The following example shows a definition of dual (IPv4 and IPv6) redirect action for the next-hop mode in the default forward class.

```
[local]Redback(config)#forward policy Demo
 redirect destination ip next-hop 3.4.5.6
 redirect destination ipv6 next-hop 2001:0f68:::0202:B3FF:FE1E:8329 default
  ip access-group Acl-demo-ipv4 local
  ipv6 access-group Acl-demo-ipv6 local
   class http-ipv4
    redirect destination ip next-hop 10.1.1.1
   class http-ipv6
    redirect destination ipv6 next-hop 2001:0f68:::0202:B3FF:FE1E:3333
```

3.3.2 Example IPv4 and IPv6 Redirection Using ACLs

The following example shows a definition of dual (IPv4 and IPv6) redirect action the IPv4 and IPv6 Policy ACLs each with a single permit `any any` rule that matches corresponding “all” IPv4 and IPv6 forward classes without need to define dual redirect action in the default class..

```
[local]Redback(config-ctx)#policy access-list Acl-demo-ipv4
  seq 10 permit ip any any class ipv4all
!
ipv6 policy access-list Acl-demo-ipv6
  seq 10 permit ipv6 any any class ipv6all
!
forward policy Demo
 ip access-group Acl-demo-ipv4 local
 ipv6 access-group Acl-demo-ipv6 local
  class ipv4all
   redirect destination ip next-hop 10.1.1.1
  class ipv6all
   redirect destination ipv6 next-hop 2001:0f68:::0202:B3FF:FE1E:3333
```

3.3.3 Example IPv6 Redirection Over Default Route

The following example redirects traffic to the next-hop IP address, 192.1.1.1. If that address follows the default route, the SmartEdge routerSM family chassis first attempts to redirect traffic to the next-hop IP address, 10.1.1.1. If that address follows the default route, then as last resort, the SmartEdge routerSM also forwards the traffic over the default route.

```
[local]Redback#config
[local]Redback(config)#forward policy RedirectPolicy
[local]Redback(config-policy-frwd)#redirect destination ip next-hop 192.1.1.1 10.1.1.1
```

The following examples show a redirect of IPv6 traffic received on the interface i22 over the default route on the interface i26.

First a forward policy for IPv6 traffic is configured with an unreachable destination address:

```
[local]Redback#config
[local]Redback(config)#forward policy Demo
 [local]Redback(config-policy-frwd)#redirect destination ip next-hop 3.5.0.4 default 2.4.0.2
 [local]Redback(config-policy-frwd)#redirect destination ipv6 next-hop 3:5::5
```

Configure the interface i26 to be the IPv6 default route:

```
[local]Redback(config-ctx)#ipv6 route 0:0::0/0 2:6::1
[local]Redback(config-ctx)#interface i26
[local]Redback(config-if)#ipv6 address 2:6::1/64
[local]Redback(config-if)#commit
```

Use the **`show ipv6 route`** command to verify the settings.

```
[local]Redback(config-dot1q-pvc)#show ipv6 route
 Gateway of last resort is 2:6::1 to network 0::0

Type    Network             Next Hop        Dist    Metric    UpTime
Interface

> S     ::/0                2:6::1             1         0  18:21:43  i26
> C     2:2::/64                               0         0  18:21:56  i22
> C     2:3::/64                               0         0  18:21:56  i23
 > C     2:6::/64                               0         0  18:21:56  i26
```

Bind the forward policy to the ingress interface i22 and the egress interface i26 to VLAN 4:

```
port ethernet 2/2
 no shutdown
 bind interface i22 local
 forward policy Demo in
!
port ethernet 2/3
 no shutdown
 encapsulation dot1q
 dot1q pvc 1
  bind interface i23 local
 dot1q pvc 4
  bind interface i26 local
```

IPv6 traffic received on the interface i22 will be redirected over the default route on the interface i26.

3.4 Traffic Drop

The following example show how to implement traffic dropping for:

- ICMP traffic from host 51.1.1.2
- PIM packets

This configuration allows all other traffic flow in the normal path.

Traffic comes in through the interface, **incoming_traffic**, and leaves the router through the interface, **normal_traffic**. Figure 3 displays the network topology for this example.

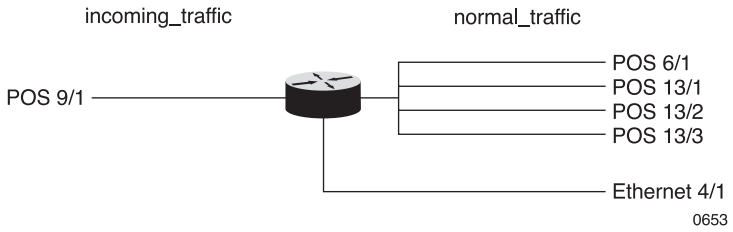

Figure 3 Basic Traffic Drop Network Topology (653)

The following example shows how to configure the interface.

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#interface e1
[local]Redback(config-if)#ip address 31.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface incoming_traffic
[local]Redback(config-if)#ip address 51.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface normal_traffic
[local]Redback(config-if)#ip address 41.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface p1
[local]Redback(config-if)#ip address 21.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface p2
[local]Redback(config-if)#ip address 22.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface p3
[local]Redback(config-if)#ip address 23.1.1.1/24
```

The following example shows how to configure the policy ACL.

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#policy access-list PBR_Drop_ACL
[local]Redback(config-access-list)#seq 10 permit icmp host 51.1.1.2 class ICMP
[local]Redback(config-access-list)#seq 20 permit pim any class PIM
```

The following example shows how to configure the forward policy.

```
[local]Redback(config)#forward policy DropPolicy
[local]Redback(config-policy-frwd)#ip access-group PBR_Drop_ACL local
[local]Redback(config-policy-group)#class ICMP
[local]Redback(config-policy-group-class)#drop
[local]Redback(config-policy-group-class)#exit
[local]Redback(config-policy-group)#class PIM
[local]Redback(config-policy-group-class)#drop
```

The following configuration shows how to attach the forward policy to an incoming circuit and bind interfaces to output ports:

```
[local]Redback(config)#port ethernet 4/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface e1 local
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 6/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface normal_traffic local
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 9/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface incoming_traffic local
[local]Redback(config-port)#forward policy DropPolicy in
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 13/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface p1 local
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 13/2
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface p2 local
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 13/3
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface p3 local
```

3.5 Combination of Traffic Mirror, Redirect, and Drop in One Policy

The following example shows how to implement the following functions:

- Redirect all web traffic to 100.1.1.2
- Mirror all forwarded UDP traffic to POS port 13/2
- Mirror all dropped IP packets to Ethernet port 4/1 not more frequently than once every three seconds
- Drop all ICMP traffic from 50.1.1.2
- Drop all PIM traffic
- Mirror all other traffic to POS port 13/3

Traffic comes in through the interface, **incoming_traffic**, and leaves the box through the interface, **normal_traffic**. Figure 4 displays the network topology for the configuration example with traffic mirroring, redirect, and drop conditions in one policy.

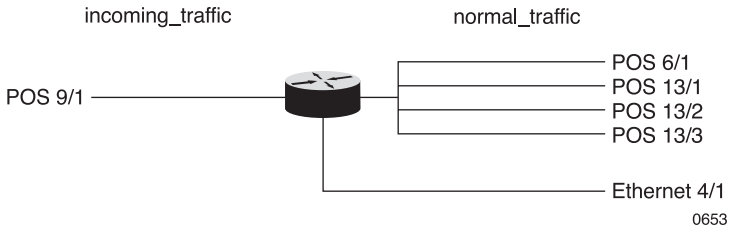

Figure 4 Basic Network Topology for Mirroring, Redirect, and Drop in One Policy (653)

The following example shows how to configure the interface.

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#interface e1
[local]Redback(config-if)#ip address 31.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface incoming_traffic
[local]Redback(config-if)#ip address 51.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface normal_traffic
[local]Redback(config-if)#ip address 41.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface p1
[local]Redback(config-if)#ip address 21.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface p2
[local]Redback(config-if)#ip address 22.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface p3
[local]Redback(config-if)#ip address 23.1.1.1/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#ip route 100.1.1.0/24 21.1.1.2
```

The following example shows how to configure the policy ACL.

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#policy access-list PBR_ACL
[local]Redback(config-access-list)#seq 10 permit tcp any eq www any class WEB
[local]Redback(config-access-list)#seq 20 permit tcp any any eq www class WEB
[local]Redback(config-access-list)#seq 30 permit udp any class UDP
[local]Redback(config-access-list)#seq 40 permit icmp host 50.1.1.2 class ICMP
[local]Redback(config-access-list)#seq 50 permit pim any class PIM
[local]Redback(config-access-list)#seq 60 permit ip any class IP
```

The following example shows how to configure the forward policy.

```
[local]Redback(config)#forward policy GeneralPolicy
[local]Redback(config-policy-frwd)#mirror destination DroppedTraffic dropped sampling 3000
[local]Redback(config-policy-frwd)#ip access-group PBR_ACL local
[local]Redback(config-policy-group)#class WEB
[local]Redback(config-policy-group-class)#redirect destination ip next-hop 100.1.1.2
[local]Redback(config-policy-group-class)#exit
[local]Redback(config-policy-group)#class UDP
[local]Redback(config-policy-group-class)#mirror destination UdpTraffic forwarded
[local]Redback(config-policy-group-class)#exit
[local]Redback(config-policy-group)#class ICMP
[local]Redback(config-policy-group-class)#drop
[local]Redback(config-policy-group-class)#exit
[local]Redback(config-policy-group)#class PIM
[local]Redback(config-policy-group-class)#drop
[local]Redback(config-policy-group-class)#exit
[local]Redback(config-policy-group)#class IP
[local]Redback(config-policy-group-class)#mirror destination IpTraffic all
```

The following configuration shows how to apply the policy to an incoming circuit and define the output destinations:

```
[local]Redback(config)#port ethernet 4/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface e1 local
[local]Redback(config-port)#forward output DroppedTraffic
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 6/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface normal_traffic local
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 9/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface incoming_traffic local
[local]Redback(config-port)#forward policy GeneralPolicy in
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 13/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface p1 local
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 13/2
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface p2 local
[local]Redback(config-port)#forward output UdpTraffic
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 13/3
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface p3 local
[local]Redback(config-port)#forward output IpTraffic
```

3.6 Configure IPv4 and IPv6 Policy ACLs and Associate Them With a Forward Policy

The following example shows how to configure and apply IPv4 and IPv6 policy ACLs for forward policy classification.

Note: Forward policies currently for IPv6 packets support only: local (HTTP) or next-hop redirect and drop. Because HTTP redirect actions are only supported for subscriber circuits, there is no benefit to applying a forward policy to a nonsubscriber circuit that carries only IPv6 traffic.

```
[local]Redback(config)#context isp
[local]Redback(config-ctx)#!
[local]Redback(config-ctx)#interface subs multiband
[local]Redback(config-if)#ip address 17.1.1.1/21
[local]Redback(config-if)#ipv6 address 2001:a:b::1/48
[local]Redback(config-if)#ip pool 17.1.1.0/24
[local]Redback(config-if)#!
[local]Redback(config-if)#aaa authentication subscriber radius
[local]Redback(config-ctx)#!
[local]Redback(config-ctx)#policy access-list find_tcp
[local]Redback(config-access-list)#seq 10 permit tcp any any class tcp
[local]Redback(config-access-list)#!
[local]Redback(config-access-list)#ipv6 policy access-list find_tcp
[local]Redback(config-ipv6-access-list)#seq 10 permit tcp any any class tcp
[local]Redback(config-ipv6-access-list)#!
[local]Redback(config-ipv6-access-list)#http-redirect profile h_redirect
!
[local]Redback(config-hr-profile)#url "http://2.2.2.2:8888"
[local]Redback(config-hr-profile)#ipv6 url "http://[2000:1:2::1]:80"
[local]Redback(config-hr-profile)# message "to be redirected to this address"
[local]Redback(config-hr-profile)#!
[local]Redback(config-hr-profile)#subscriber default
[local]Redback(config-sub)#ip address pool
[local]Redback(config-sub)#http-redirect profile h_redirect
[local]Redback(config-sub)#forward policy red_pol in
[local]Redback(config-sub)#access-list count ip ipv6 ipv6-policy policy
[local]Redback(config-sub)#!
[local]Redback(config-sub)#! ** End Context **
[local]Redback(config-sub)#!
[local]Redback(config-sub)#http-redirect server
[local]Redback(config-hr-server)#port 80 8888
[local]Redback(config-hr-server)#!
[local]Redback(config-hr-server)#forward policy red_pol
!
[local]Redback(config-policy-frwd)#ip access-group find_tcp isp
[local]Redback(config-policy-group)#ipv6 access-group find_tcp isp
[local]Redback(config-policy-group)#class tcp
[local]Redback(config-policy-group-class)#redirect destination local
[local]Redback(config-policy-group-class)#
```

```
[local]Redback(config-policy-group-class)#
[local]Redback(config-policy-group-class)#end
!
```

3.7 Redirect Interactions with ICMPv6

IPv6 nodes on the same link use the Neighbor Discovery (ND) protocol to discover each other's presence. To allow ND notifications, exclude the incoming ND messages Router Solicitation (133), Neighbor Solicitation (135) and Redirect (137) from redirects. However, for a proper 'ping' operation, redirect the ICMPv6 Echo Request (128) message. Redirect approaches for ICMPv6 messages are:

- Select only non-ICMPv6 traffic for redirect action.
- Use a distinct rule set for specific ICMPv6 messages in the IPv6 Policy ACL that matches a corresponding "exclude" (no action) IPv6 forwarding class.

The following example redirects www traffic only. ICMPv6 traffic does not match any rule and is subject to the default class, which has no redirect action. In this example, ICMPv6 errors for www traffic might take a different route back to the destination.

```
[local]Redback#
!
ipv6 policy access-list www-redirect-only
seq 10 permit tcp any any eq www class HTTP
!
forward policy Test
ipv6 access-group www-redirect-only local
class HTTP
redirect destination ipv6 next-hop 2:3::2
```

This example shows an explicit exclusion list for ND messages. The default class has no action configured and serves as the exclusion class. The forward policy redirects all IPv6 traffic, except for the ICMPv6 traffic required for a proper ND operation.

```
[local]Redback#
ipv6 policy access-list Acl-exclude-demo-ipv6
seq 10 permit icmp any any icmp-type nd router-solicitation traffic-class eq df class default
seq 20 permit icmp any any icmp-type nd neighbor-solicitation traffic-class
eq df class default
seq 30 permit icmp any any icmp-type nd redirect traffic-class eq df class default
seq 40 permit ipv6 any any class other-ipv6
!
forward policy Demo
ipv6 access-group Acl-exclude-demo-ipv6 local
class other-ipv6
redirect destination ipv6 next-hop 2:4::2 2:5::2
```