Policy-based Routing Configuration Guide, 17.2.0

November 2017

Supporting AT&T Vyatta Network Operating System

Contents

About This Guide 6

Policy-based routing 7

Introduction 7

Defining a routing policy 7

Routing policy rules 7

PBR behavior 8

Packet forwarding path 8

Per packet logging 8

Deep Packet Inspection 8

Configuration Examples 10

PBR routing example 10

Binding interfaces to PBR tables 12

DPI PBR examples 14

Policy-Based Routing Commands 17

clear policy 17

interfaces bonding <dpFbondN> policy route pbr <name> 17

interfaces bonding <dpFbondN> vif <vif-id> policy route pbr <name> 18

interfaces dataplane interface policy route pbr name 18

interfaces dataplane <dpxx> vif <vif-id> policy route pbr <name> 19

interfaces l2tpeth <lttpN> policy route pbr <name> 20

interfaces l2tpeth <lttpN> vif <vif-id> policy route pbr <name> 20

interfaces openvpn <vtunx> policy route pbr <name> 21

interfaces tunnel <tunx> policy route pbr <name> 22

interfaces vti <vtix> policy route pbr <name> 22

policy route pbr name rule 23

policy route pbr name rule action 24

policy route pbr <name> rule <rule-number> application name <name> 25

policy route pbr <name> rule <rule-number> application type <type> 25

policy route pbr name rule af 26

policy route pbr name rule description 27

policy route pbr name rule destination 28

policy route pbr name rule disable 29

policy route pbr name rule icmp 30

policy route pbr name rule icmpv6 31

policy route pbr <name> rule <number> ipv6-route type <number> 32

policy route pbr name rule log 33

policy route pbr <pbr-policy> rule <rule> path-monitor monitor <path-monitor-monitor-name> policy <path-monitor-policy-name> 34

policy route pbr name rule port 35

policy route pbr <name> rule <number> pcp <value> 36

policy route pbr name rule protocol 37

policy route pbr name rule source address ... 38
policy route pbr <name> rule <rule-number> source mac-address <address> ... 39
policy route pbr name rule source port ... 40
policy route pbr name rule table ... 41
policy route pbr name rule tcp flags ... 42
show application name <name> ... 43
show application type <type> ... 43
show policy route ... 44
show policy route table ... 45
Related commands ... 45
ICMP Types ... 47
ICMPv6 Types ... 50
Supported Interface Types ... 52
VRF support for PBR ... 55
Command support for VRF routing instances ... 55
Configuring policy-based routing on a routing instance ... 57
List of Acronyms ... 60

Copyright Statement

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T.

About This Guide

This guide describes how to define and configure routing policies on AT&T products that run on the AT&T Vyatta Network Operating System (referred to as a virtual router, vRouter, or router in the guide).

Policy-based Routing

Introduction

Policy-based routing (PBR) enables you to use IP traffic rules to classify traffic based on its attributes and apply processing differentially according to the classification, and to selectively route IP packets, for example, to an alternate next hop. PBR on the AT&T Vyatta vRouter is supported on incoming Layer 3, Layer 4, and Layer 7 traffic.

Only packets that pass through the firewall (if any) are considered for policy-based routing provided the interface is assigned a routing policy.

When no routing policies are applied, routing decisions are made by using the default (main) routing table (Table 254) of the system.

PBR policies can be applied to bonding, bonding VIF (virtual interface), bridge (vCPE only), dataplane, dataplane VIF, L2TPv3, L2TPv3 VIF, OpenVPN, tunnel, and VTI (Virtual Tunnel Interface) interfaces for inbound traffic, but not to loopback interfaces.

Note:

A performance drop is expected when PBR is configured on an interface.

On the AT&T Vyatta vRouter, you cannot apply policy-based routing to locally generated packets.

Defining a routing policy

The routing policy classifies traffic and specifies the handling that should take place for different classes. This classification and handling are accomplished by using a set of policy rules.

Rules are configured with match criteria that include an extensive set of attributes—including protocol, source and destination addresses and ports, fragmentation, ICMP or ICMPv6 type, and TCP flags. You can also preconfigure groups of addresses, ports, and networks and refer to these groups in policy rules.

The routing policy must be applied to an interface for the policy to be effective.

To implement policy-based routing, perform the following steps.

1. Define the policy rules.
2. Attach the policy to an ingress interface.
3. Create a route in a PBR table other than Table 254.

Info:

Note: Table 254 is also known as the main table.

Routing policy rules

Packets that match the PBR rule criteria are subject to either of the following actions.

- They are routed by using a specific PBR routing table.
- They are dropped (if the **drop** action is set).

Packets that match the rule parameters are considered for policy-based routing. As many as 9,999 rules in a policy are supported. If no match criteria are specified, all packets are routed according to the default Table 254.

The packets that do not match any policy rule are routed according to the routes in the main table.

Note: You can configure rules to match IPv4 ICMP, IPv6 ICMP, IPv6 routing header, or TCP without specifying the respective protocol, provided that a protocol specific match option is present. For example TCP flags, ICMP type.

Routing policy rules are executed in numeric sequence, from lowest to highest.

Note: To avoid having to renumber routing policy rules, a good practice is to number rules in increments of 10. This increment allows room for the insertion of new rules within the policy.

PBR behavior

Routes remain persistent in the controller. If the data plane goes down and up, the routes are automatically re-established without the need for reconfiguration.

PBR rules can be changed dynamically and do not require the rebinding of the PBR policy to an interface.

Configurations for VLAN-based classification, MAC address, and packet mangling are not supported.

The controller automatically continuously resynchronizes the route information to the data plane.

Multiple PBR policies can be applied to an interface. For best results, we recommend that the rules in each policy are unique.

Packet forwarding path

When enabled, PBR processes incoming packets after packet validation and firewall action. Packets received by the data plane ingress interfaces for transmission to the egress interface follow the forwarding path listed below. PBR operates on the VRF of the interface that the policy is applied to.

1. Packet validation and reassembly
2. Firewall
3. DNAT
4. PBR classification, route table ID determination
5. SNAT
6. Firewall
7. QoS
8. Transmit out of an egress interface

Info:

Per packet logging

You can configure the vRouter to log every packet that matches a network packet filter rule.

Note: Per packet logging generates large amounts of output and can negatively affect the performance of the entire system. Use per packet logging only for debugging purposes.

When logging is enabled, all log messages appear in the /var/log/dataplane/vplane.log file. This file is rotated and compressed daily, and the last seven log files are automatically maintained by the system.

AT&T recommends limiting per packet logging to debugging. Per packet logging occurs in the forwarding paths and can greatly reduce the throughput of the system and dramatically increase the disk space used for the log files.

To implement per packet logging for debugging purposes, include the **log** keyword when specifying a rule. When the logging option is specified, a log message containing the parameters of the packet is generated and logged.

Deep Packet Inspection

Deep packet inspection (DPI) is a packet filtering process that examines the contents of packets. DPI identifies different types of packets by application such as VOIP, email, web so that different actions can be taken on the packets. These actions include traffic management and blocking. DPI is available for Policy-Based Routing (PBR) and Quality of Service (QoS). For more information on QoS refer to AT&T Vyatta Network Operating System QoS Configuration Guide. DPI is configured to classify and route traffic by using the new application name or application type CLI. The application name and application type configurations are mutually exclusive. You

can use only one configuration at a time within a single rule. However, you can configure different application names and application types in separate rules. A single application can be matched from a list of DPI engine applications at the most granular level.

Configuration Examples

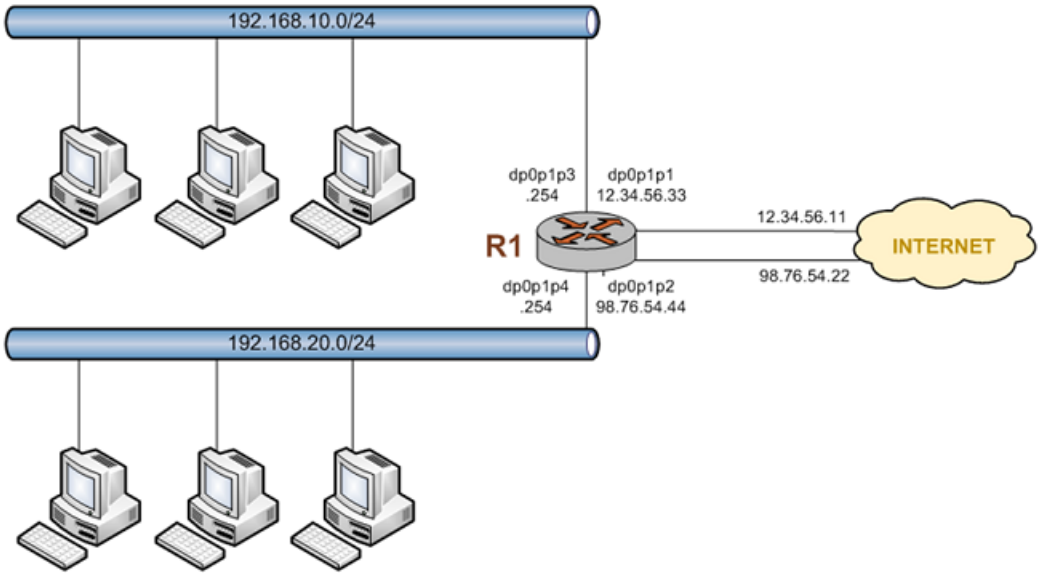
PBR routing example

The following figure shows a simple site that uses PBR on the AT&T Vyatta vRouter (R1) to route traffic from two different internal subnets to two Internet links.

The following conditions apply to this scenario:

- All Internet-bound traffic from subnet 192.168.10.0/24 is routed out interface dp0p1p1.
- All Internet-bound traffic from subnet 192.168.20.0/24 is routed out interface dp0p1p2.

Figure 1: Routing using PBR



To configure the scenario, perform the following steps in configuration mode.

Table 1: Routing using PBR

Step	Command
Create Rule 10.	`vyatta@R1# set policy route pbr myroute rule 10 address-family ipv4` `vyatta@R1# set policy route pbr myroute rule 10 action accept`
Specify the source address to match. In this case, any address on subnet 192.168.10.0/24 is a match.	`vyatta@R1# set policy route pbr myroute rule 10 source address 192.168.10.0/24`
Specify that all matching packets use alternate routing table 1.	`vyatta@R1# set policy route pbr myroute rule 10 table 1`

Step	Command

Create rule 20.	`vyatta@R1# set policy route pbr myroute rule 20 address-family ipv4`
	
	`vyatta@R1# set policy route pbr myroute rule 20 action accept`

Specify the source address to match. In this case, any address on subnet 192.168.20.0/24 is a match.	`vyatta@R1# set policy route pbr myroute rule 20 source address 192.168.20.0/24`
Specify that all matching packets use alternate routing table 2.	`vyatta@R1# set policy route pbr myroute rule 20 table 2`
Commit the changes.	`vyatta@R1# commit`

Show the policy-based routing configuration.	`vyatta@R1# show policy route`
	`route {`
	`  pbr myroute {`
	`  rule 10 {`
	`  action accept`
	`  address-family ipv4`
	`  source {`
	`  address 192.168.10.0/24`
	`  }`
	`  table 1`
	`  }`
	`  rule 20 {`
	`  action accept`
	`  address-family ipv4`
	`  source {`
	`  address 192.168.20.0/24`
	`  }`
	`  table 2`
	`  }`
	`  }`
	`}`

Create the alternative routing table 1.	`vyatta@R1# set protocols static table 1 route 12.34.56.0/24 next-hop 12.34.56.11`
Create the alternative routing table 2.	`vyatta@R1# set protocols static table 2 route 98.76.54.0/24 next-hop 98.76.54.22`
Commit the change.	`vyatta@R1# commit`

Step	Command

Show the alternate routing table configuration.	`vyatta@R1# show protocols static`
	
	`static {`
	`  table 1 {`
	`  route 12.34.56.0/24 {`
	`  next-hop 12.34.56.11`
	`  }`
	`  }`
	`  table 2 {`
	`  route 98.76.54.0/24 {`
	`  next-hop 98.76.54.22`
	`  }`
	`  }`
	`  }`

Apply the IP addresses to the corresponding data plane interfaces.	`vyatta@R1# set interfaces dataplane dp0p1p1`
	`address 12.34.56.33/24`
	`vyatta@R1# set interfaces dataplane dp0p1p2`
	`address 98.76.54.44/24`
	`vyatta@R1# set interfaces dataplane dp0p1p3`
	`address 192.168.10.254/24`
	`vyatta@R1# set interfaces dataplane dp0p1p4`
	`address 192.168.20.254/24`

Apply the policy route with dp0p1p3, and dp0p1p4 interfaces	`vyatta@R1# set interfaces dataplane dp0p1p3`
	`policy route pbr myroute`
	`vyatta@R1# set interfaces dataplane dp0p1p4`
	`policy route pbr myroute`

Show the data plane interface configuration.	`vyatta@R1# show interfaces dataplane`
	
	`dataplane dp0p1p1 {`
	`  address 12.34.56.33/24`
	`}`
	`dataplane dp0p1p2 {`
	`  address 98.76.54.44/24`
	`}`
	`dataplane dp0p1p3 {`
	`  address 192.168.10.254/24`
	`  policy {`
	`  route {`
	`  pbr myroute`
	`  }`
	`  }`
	`}`
	`dataplane dp0p1p4 {`
	`  address 192.168.20.254/24`
	`  policy {`
	`  route {`
	`  pbr myroute`
	`  }`
	`  }`
	`}`

Binding interfaces to PBR tables

To configure an interface-based static route in a policy route table, perform the following steps:

Table 2: Applying a policy route to an interface

Step	Command
Configure the interface route for the interface.	`vyatta@R1# set protocols static table 10 interface-route 192.168.20.254/24 nexthop-interface dp0p256p1 distance 25`

View the configuration.	`vyatta@vyatta:~$ show protocols`
	
	`protocols {`
	`  static {`
	`  table 10 {`
	`  interface-route 192.168.20.254/24 {`
	`  nexthop-interface dp0p256p1 {`
	`  distance 25`
	`  }`
	`  }`
	`  }`
	`  }`
	`}`

Commit the change.	`vyatta@R1# commit`

DPI PBR examples

The following sections describe how to forward video chat traffic in a policy route table, send VPN tunnel traffic into a corporate VRF, and block a specific type of application.

Forwarding video traffic

To forward video chat traffic in a policy route table, perform the following steps:

Table 3: Forwarding video chat traffic in a policy route table

Step	Command
Create rule 10 to accept IPv4 traffic.	`vyatta@R1# set policy route pbr myroute1 rule 10 action accept` `vyatta@R1# set policy route pbr myroute1 rule 10 address-family 'ipv4'`
Specify the application type	`vyatta@R1# set policy route pbr myroute1 rule 10 application type video_chat`
Specify that all matching packets use alternate routing table 2.	`vyatta@R1# set policy route pbr myroute1 rule 10 table 2`
Specify the routing instance to route the traffic into.	`vyatta@R1# set policy route pbr myroute1 rule 10 routing-instance corporate`
Commit the change.	`vyatta@R1# commit`

Show the traffic forwarding configuration.	`show policy route pbr`
	`pbr myroute1 {`
	`  rule 10 {`
	`  action accept`
	`  address-family ipv4`
	`  application {`
	`  type video_chat`
	`  }`
	`  table 2`
	`  }`
	`}`

Sending VPN tunnel traffic into the corporate VRF

To send VPN tunnel traffic into a VRF and then use the VRF's default routing table, perform the following steps:

Table 4: Sending VPN tunnel traffic into a VRF

Step	Command
Create rule 10 to accept IPv4 traffic.	`vyatta@R1# set policy route pbr myroute2 rule 10 action accept` `vyatta@R1# set policy route pbr myroute2 rule 10 address-family 'ipv4'`

Step	Command
Specify the application type.	`vyatta@R1# set policy route pbr myroute2 rule 10 application type vpn_tun`
Specify the routing instance to route the traffic into.	`vyatta@R1# set policy route pbr myroute2 rule 10 routing-instance corporate`
Commit the change.	`vyatta@R1# commit`

Show the traffic forwarding configuration.	`show policy route pbr`
	`pbr myroute2 {`
	`  rule 10 {`
	`  action accept`
	`  address-family ipv4`
	`  application {`
	`  type vpn_tun`
	`  }`
	`  routing-instance corporate`
	`  }`
	`}`

Note:

Since no table is configured, the routing instance's default routing table is used.

Blocking a specific type of application

To block YouTube traffic, perform the following steps:

Table 5: Blocking YouTube traffic

Step	Command
Create rule 10 to drop IPv4 traffic.	`vyatta@R1# set policy route pbr myroute3 rule 10 action drop` `vyatta@R1# set policy route pbr myroute3 rule 10 address-family 'ipv4'`
Specify the application name.	`vyatta@R1# set policy route pbr myroute3 rule 10 application name youtube`
Commit the change.	`vyatta@R1# commit`

Step	Command

Show the traffic forwarding configuration.	`show policy route pbr`
	`pbr myroute3 {`
	`  rule 10 {`
	`  action drop`
	`  address-family ipv4`
	`  application {`
	`  name youtube`
	`  }`
	`  }`
	`}`

Policy-based Routing Commands

clear policy

Clears the statistics for route policies.

Syntax:
```
clear policy
```

Operational mode

Use this command to clear the statistics for policy-based routing.

interfaces bonding <dpFbondN> policy route pbr <name>

Applies a PBR policy to an Ethernet link bond group.

Syntax:
```
set interfaces bonding dpFbondN policy route pbr name
```

Syntax:
```
delete interfaces bonding dpFbondN policy route pbr [ name ]
```

Syntax:
```
show interfaces bonding dpFbondN policy route pbr [ name ]
```

dpFbondN
The identifier for a bond group. The identifier ranges from dp0bond0 through dp0bond99.

name
The name of a PBR policy.

Configuration mode

```
interfaces bonding {
    policy {
        route {
            pbr name
        }
    }
}
```

A routing policy has no effect on traffic traversing the system until it has been applied to an interface.

To use the policy-based routing feature, you must define a routing policy by using the `set policy route pbr` *name* **`rule`** *number* command, and then apply the routing policy to interfaces by using the `interfaces bonding` *dpFbondN* **`policy route pbr`** *name* command. Once applied, the rule set acts as a packet filter.

Use the `set` form of this command to apply a PBR policy to an interface.

Use the `delete` form of this command to remove a PBR policy, or all PBR policies, from an interface.

Use the `show` form of this command to display a PBR policy configuration, or all PBR policy configurations, for an interface.

interfaces bonding <dpFbondN> vif <vif-id> policy route pbr <name>

Applies a PBR policy to a virtual interface of an Ethernet link bond group.

Syntax:
`set interfaces bonding` *dpFbondN* **`vif`** *vif-id* **`policy route pbr`** *name*

Syntax:
`delete interfaces bonding` *dpFbondN* **`vif`** *vif-id* **`policy route pbr`** [*name*]

Syntax:
`show interfaces bonding` *dpFbondN* **`vif`** *vif-id* **`policy route pbr`** [*name*]

dpFbondN
The identifier for a bond group. The identifier ranges from dp0bond0 through dp0bond99.

vif-id
A virtual interface (vif) ID. The ID ranges from 1 through 99999.

name
The name of a PBR policy.

Configuration mode

```
interfaces {
        bonding interface-name {
                vif vif-id {
                        policy {
                                pbr policy-name
                        }
                }
        }
}
```

A routing policy has no effect on traffic traversing the system until it has been applied to an interface.

To use the policy-based routing feature, you must define a routing policy by using the `set policy route pbr` *name* **`rule`** *number* command, and then apply the routing policy to interfaces by using the `interfaces bonding` *dpFbondN* **`vif`** *vif-id* **`policy route pbr`** *name* command. Once applied, the rule set acts as a packet filter.

Use the `set` form of this command to apply a PBR policy to an interface.

Use the `delete` form of this command to remove a PBR policy, or all PBR policies, from an interface.

Use the `show` form of this command to display a PBR policy configuration, or all PBR policy configurations, for an interface.

interfaces dataplane <dpxx> policy route pbr <name>

Applies a PBR policy to inbound traffic on a data plane interface.

Syntax:
`set interfaces dataplane` *dpxx* **`policy route pbr`** *name*

Syntax:
`delete interfaces dataplane` *dpxx* **`policy route pbr`** [*name*]

Syntax:
`show interfaces dataplane` *dpxx* **`policy route pbr`** [*name*]

dpxx
The name of a data plane interface, where dpx specifies the data plane identifier (ID). Currently, only dp0 is supported.

name
The name of a PBR policy.

Configuration mode

```
interfaces dataplane interface {
    policy {
        route {
            pbr name
        }
    }
}
```

A routing policy has no effect on traffic traversing the system until it has been applied to an interface.

To use the policy-based routing feature, you must define a routing policy by using the `set policy route pbr` *name* **`rule`** *number* command, and then apply the routing policy to interfaces by using the `interfaces dataplane` *dpxx* **`policy route pbr`** *name* command. Once applied, the rule set acts as a packet filter.

Use the `set` form of this command to apply a PBR policy to an interface.

Use the `delete` form of this command to remove a PBR policy, or all PBR policies, from an interface.

Use the `show` form of this command to display a PBR policy configuration, or all PBR policy configurations, for an interface.

interfaces dataplane <dpxx> vif <vif-id> policy route pbr <name>

Applies a PBR policy to a virtual interface of a data plane.

Syntax:
`set interfaces dataplane` *dpxx* **`vif`** *vif-id* **`policy route pbr`** *name*

Syntax:
`delete interfaces dataplane` *dpxx* **`vif`** *vif-id* **`policy route pbr`** [*name*]

Syntax:
`show interfaces dataplane` *dpxx* **`vif`** *vif-id* **`policy route pbr`** [*name*]

dpxx
The name of a data plane interface, where dpx specifies the data plane identifier (ID). Currently, only dp0 is supported.

vif-id
A virtual interface (vif) ID. The ID ranges from 1 through 99999.

name
The name of a PBR policy.

Configuration mode

```
interfaces {
        dataplane interface-name {
                vif vif-id {
                        policy {
                                pbr policy-name
                        }
                }
        }
```

```
}
```

A routing policy has no effect on traffic traversing the system until it has been applied to an interface.

To use the policy-based routing feature, you must define a routing policy by using the `set policy route pbr` *name* **`rule`** *number* command, and then apply the routing policy to interfaces by using the `set interfaces dataplane` *dpxx* **`vif`** *vif-id* **`policy route pbr`** *name* command. Once applied, the rule set acts as a packet filter.

Use the `set` form of this command to apply a PBR policy to an interface.

Use the `delete` form of this command to remove a PBR policy, or all PBR policies, from an interface.

Use the `show` form of this command to display a PBR policy configuration, or all PBR policy configurations, for an interface.

interfaces l2tpeth <lttpN> policy route pbr <name>

Applies a PBR policy to inbound traffic on an L2TPv3 static tunnel interface.

Syntax:

```
set interfaces l2tpeth lttpN policy route pbr name
```

Syntax:

```
delete interfaces l2tpeth lttpN policy route pbr [ name ]
```

Syntax:

```
show interfaces l2tpeth lttpN policy route pbr [ name ]
```

lttpN

L2TPv3 static L2TPv3 tunnel interface. The interface ranges from lttp0 through lttpN, where N is a nonnegative integer.

name

The name of a PBR policy.

Configuration mode

```
interfaces l2tpeth {
    policy {
        route {
            pbr name
        }
    }
}
```

A routing policy has no effect on traffic traversing the system until it has been applied to an interface.

To use the policy-based routing feature, you must define a routing policy by using the `set policy route pbr` *name* **`rule`** *number* command, and then apply the routing policy to interfaces by using the `interfaces l2tpeth` *lttpN* **`policy route pbr`** *name* command. Once applied, the rule set acts as a packet filter.

Use the `set` form of this command to apply a PBR policy to an interface.

Use the `delete` form of this command to remove a PBR policy, or all PBR policies, from an interface.

Use the `show` form of this command to display a PBR policy configuration, or all PBR policy configurations, for an interface.

interfaces l2tpeth <lttpN> vif <vif-id> policy route pbr <name>

Applies a PBR policy to inbound traffic on a virtual interface of an L2TPv3 static tunnel.

Syntax:

```
set interfaces l2tpeth lttpN vif vif-id policy route pbr name
```

Syntax:

```
delete interfaces l2tpeth lttpN vif vif-id policy route pbr [ name ]
```

Syntax:

```
show interfaces l2tpeth lttpN vif vif-id policy route pbr [ name ]
```

vif-id

A virtual interface (vif) ID. The ID ranges from 1 through 4094.

name

The name of a PBR policy.

Configuration mode

```
interfaces {
        l2tpeth interface-name {
                 vif vif-id {
                        policy {
                                pbr policy-name
                        }
                }
        }
}
```

A routing policy has no effect on traffic traversing the system until it has been applied to an interface.

To use the policy-based routing feature, you must define a routing policy by using the `set policy route pbr` *name* **`rule`** *number* command, and then apply the routing policy to interfaces by using the `interfaces l2tpeth` *lttpN* **`vif`** *vif-id* **`policy route pbr`** *name* command. Once applied, the rule set acts as a packet filter.

Use the `set` form of this command to apply a PBR policy to an interface.

Use the `delete` form of this command to remove a PBR policy, or all PBR policies, from an interface.

Use the `show` form of this command to display a PBR policy configuration, or all PBR policy configurations, for an interface.

interfaces openvpn <vtunx> policy route pbr <name>

Applies a PBR policy to inbound traffic on an OpenVPN tunnel interface.

Syntax:

```
set interfaces openvpn vtunx policy route pbr name
```

Syntax:

```
delete interfaces openvpn vtunx policy route pbr [ name ]
```

Syntax:

```
show interfaces openvpn vtunx policy route pbr [ name ]
```

vtunx

The identifier of an OpenVPN interface. The identifier ranges from *vtun0* through *vtunx*, where *x* is a nonnegative integer.

name

The name of a PBR policy.

Configuration mode

```
interfaces openvpn {
    policy {
        route {
            pbr name
```

```
        }
    }
}
```

A routing policy has no effect on traffic traversing the system until it has been applied to an interface.

To use the policy-based routing feature, you must define a routing policy by using the `set policy route pbr` *name* **`rule`** *number* command, and then apply the routing policy to interfaces by using the `interfaces openvpn` *vtunx* **`policy route pbr`** *name* command. Once applied, the rule set acts as a packet filter.

Use the `set` form of this command to apply a PBR policy to an interface.

Use the `delete` form of this command to remove a PBR policy, or all PBR policies, from an interface.

Use the `show` form of this command to display a PBR policy configuration, or all PBR policy configurations, for an interface.

interfaces tunnel <tunx> policy route pbr <name>

Applies a PBR policy to inbound traffic on a tunnel interface.

Syntax:

```
set interfaces tunnel tunx policy route pbr name
```

Syntax:

```
delete interfaces tunnel tunx policy route pbr [ name ]
```

Syntax:

```
show interfaces tunnel tunx policy route pbr [ name ]
```

tunx

The identifier of a tunnel interface. The identifier ranges from *tun0* through *tunx*, where *x* is a nonnegative integer.

name

The name of a PBR policy.

Configuration mode

```
interfaces tunnel {
    policy {
        route {
            pbr name
        }
    }
}
```

A routing policy has no effect on traffic traversing the system until it has been applied to an interface.

To use the policy-based routing feature, you must define a routing policy by using the `set policy route pbr` *name* **`rule`** *number* command, and then apply the routing policy to interfaces by using the `interfaces tunnel` *tunx* **`policy route pbr`** *name* command. Once applied, the rule set acts as a packet filter.

Use the `set` form of this command to apply a PBR policy to an interface.

Use the `delete` form of this command to remove a PBR policy, or all PBR policies, from an interface.

Use the `show` form of this command to display a PBR policy configuration, or all PBR policy configurations, for an interface.

interfaces vti <vtix> policy route pbr <name>

Applies a PBR policy to inbound traffic on a virtual tunnel interface.

Syntax:

```
set interfaces vti vtix policy route pbr name
```

Syntax:

```
delete interfaces vti vtix policy route pbr [ name ]
```

Syntax:

```
show interfaces vti vtix policy route pbr [ name ]
```

vtix

The identifier of a virtual tunnel interface. The identifier ranges from *vti0* through *vtix*, where *x* is a nonnegative integer.

name

The name of a PBR policy.

Configuration mode

```
interfaces vti {
    policy {
        route {
            pbr name
        }
    }
}
```

A routing policy has no effect on traffic traversing the system until it has been applied to an interface.

To use the policy-based routing feature, you must define a routing policy by using the `set policy route pbr` *name* **`rule`** *number* command, and then apply the routing policy to interfaces by using the `interfaces vti` *vtix* **`policy route pbr`** *name* command. Once applied, the rule set acts as a packet filter.

Use the `set` form of this command to apply a PBR policy to an interface.

Use the `delete` form of this command to remove a PBR policy, or all PBR policies, from an interface.

Use the `show` form of this command to display a PBR policy configuration, or all PBR policy configurations, for an interface.

policy route pbr <name> rule <rule-number>

Defines an IP routing policy rule.

Syntax:

```
set policy route pbr name rule rule-number
```

Syntax:

```
delete policy route pbr name rule [ rule-number ]
```

Syntax:

```
show policy route pbr name rule
```

name

The name of an IP routing policy.

rule-number

The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one **`rule`** configuration node.

Configuration mode

```
policy {
    route {
```

```
        pbr name {
                rule rule-number
            }
        }
    }
```

A policy identifies traffic that matches parameters and specifies which routing table to use. The table defines the route for a packet to take. A routing policy is a named collection of as many as 9,999 packet-classification rules. When applied to an interface, the policy rule classifies incoming traffic.

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

Use the `set` form of this command to create a rule.

Use the `delete` form of this command to delete an existing IP routing policy.

Use the `show` form of this command to display a rule.

policy route pbr <name> rule <rule-number> action <action>

Defines the action for an IP routing policy rule.

Syntax:

`set policy route pbr` *name* **`rule`** *rule-number* **`action`** { **`drop`** | **`accept`** }

Syntax:

`delete policy route pbr` *name* **`rule`** *rule-number* **`action`** [**`drop`** | **`accept`**]

Syntax:

`show policy route pbr` *name* **`rule`** *rule-number* **`action`**

name

The name of an IP routing policy.

rule-number

The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one **`rule`** configuration node.

action

The action for an IP routing policy. The actions for an IP routing policy are accept and drop.

`accept`

Accepts the packet.

`drop`

Drops the packet silently.

Configuration mode

```
policy {
    route {
        pbr name {
               rule rule-number {
                    action accept
                    action drop
                }
            }
        }
    }
```

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

If a rule does not explicitly drop a packet in the action, the PBR action is to accept the packet, which causes it to be sent to the specified alternate routing table for lookup and forwarding.

An applied policy can only be deleted after first removing it from an assigned interface.

Use the `set` form of this command to set the action for a rule.

Use the `delete` form of this command to remove the action for a rule.

Use the `show` form of this command to display a rule within an IP routing policy.

policy route pbr <name> rule <rule-number> application name <name>

Matches applications by name.

Syntax:

```
set policy route pbr name rule rule-number application name name
```

Syntax:

```
delete policy route pbr name rule rule-number application
```

Syntax:

```
show policy route pbr name rule rule-number application
```

pbr name

The name of a PBR policy.

rule-number

The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one `rule` configuration node.

name

The name of the application. You can configure a single application name to be matched from a list of DPI engine applications at the most granular level. For more information about DPI, refer to AT&T Vyatta Network Operating System Policy-based Routing Configuration Guide

Configuration mode

```
pbr name {
       rule rule-number {
            action action
            address-family address
            application {
                name application-name
            }
   table table-number
       }
}
```

policy route pbr <name> rule <rule-number> application type <type>

Matches applications by type.

Syntax:

```
set policy route pbr name rule rule-number application type type
```

Syntax:

```
delete policy route pbr name rule rule-number application
```

Syntax:

```
show policy route pbr name rule rule-number application
```

name

The name of an PBR policy.

rule-number

The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one `rule` configuration node.

type

The type of the application. The application type provides access to less granular groups of DPI classifications such as analytics, database, social networking. An application can have multiple application types. You can configure a single application type to be matched from a list of DPI engine application types at the most granular level. For more information about DPI, refer to AT&T Vyatta Network Operating System Policy-based Routing Configuration Guide

Configuration mode

```
pbr name {
  rule rule-number {
            action action
            address-family address
            application {
                type application-type
            }
   table table-number
        }
}
```

policy route pbr <name> rule <rule-number> address-family <address-family>

Defines the address family for an IP routing policy rule.

Syntax:

```
set policy route pbr name rule rule-number address-family [ ipv4 | ipv6 ]
```

Syntax:

```
delete policy route pbr name rule rule-number address-family [ ipv4 | ipv6 ]
```

Syntax:

```
show policy route pbr name rule rule-number address-family
```

name

The name of an IP routing policy. The policy name must be unique and must not be used with other PBR policy commands.

rule-number

The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one **`rule`** configuration node.

address-family

The address-family for an IP routing policy rule. The address-family for an IP routing policy are ipv4 and ipv6.

Configuration mode

```
policy {
    route {
        pbr name {
                rule rule-number {
                    address-family ipv4
                    address-family ipv6
                }
            }
        }
    }
```

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

Use the `set` form of this command to define the address family and routing protocol for an IP routing policy rule.

Use the `delete` form of this command to remove the address family and routing protocol for an IP routing policy rule.

Use the `show` form of this command to view the address family and routing protocol for an IP routing policy rule.

policy route pbr <name> rule <rule-number> description <description>

Provides a brief description for an IP routing policy rule.

Syntax:
`set policy route pbr` *name* **`rule`** *rule-number* **`description`** *description*

Syntax:
`delete policy route pbr` *name* **`rule`** *rule-number* **`description`**

Syntax:
`show policy route pbr` *name* **`rule`** *rule-number* **`description`**

name
The name of an IP routing policy.

rule-number
The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one `rule` configuration node.

description
A brief description for the rule. If the description contains spaces, it must be enclosed in double quotation marks (").

Configuration mode

```
policy {
    route {
        pbr name {
                rule rule-number {
                    description description
                }
            }
        }
    }
```

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

Use the `set` form of this command to provide a description for an IP routing policy rule.

Use the `delete` form of this command to remove a description for an IP routing policy rule.

Use the `show` form of this command to display a description for an IP routing policy rule.

policy route pbr <name> rule <rule-number> destination <destination>

Defines the destination address for an IP routing policy rule.

Syntax:

`set policy route pbr` *name* **`rule`** *rule-number* **`destination`** { **`address`** *address* | **`mac-address`** *mac-address* | **`port`** *port* }

Syntax:

`delete policy route pbr` *name* **`rule`** *rule-number* **`destination`** [**`address`** | **`mac-address`** | **`port`**]

Syntax:

`show policy route pbr` *name* **`rule`** *rule-number* **`destination`**

name

The name of an IP routing policy.

rule-number

The numeric identifier of a policy rule. Rule numbers determine the order in which rules are processed. Each rule must have a unique rule number. The number ranges from 1 through 9999.

You can define multiple rules by creating more than one **`rule`** configuration node.

destination

The destination address for an IP routing policy rule. The destination address can be any of the following parameters.

address

Specifies an address to match. Address formats are as follows:

address-group name: An address group that is configured with a list of addresses.

ip-address: An IPv4 address.

ip-address/prefix: An IPv4 network address, where 0.0.0.0/0 matches any network.

`!` *ip-address*: All IP addresses except the specified IPv4 address.

`!` *ip-address/prefix*: All IP addresses except the specified IPv4 network address.

ipv6-address: An IPv6 address; for example, fe80::20c:29fe:fe47:f89.

ip-address/prefix: An IPv6 network address, where ::/0 matches any network; for example, fe80::20c:29fe:fe47:f88/64.

`!` *ipv6-address*: All IP addresses except the specified IPv6 address.

`!` *ip-address/prefix*: All IP addresses except the specified IPv6 network address.

mac-address

Specifies a media access control (MAC) address to match. The address format is six 8-bit numbers, separated by colons, in hexadecimal; for example, 00:0a:59:9a:f2:ba.

Note: For policy based routing, the usefulness of this parameter is limited because the MAC address is on a local interface.

port

Specifies a port to match. Port formats are as follows:

- *port-group name*: A port group that is configured with a list of ports.

- *port name*: A port name as shown in `/etc/services`, for example, http.
- *1-65535*: A port number in the range from 1 through 65535.
- *start-end*: A range of port numbers, for example, 1001-1005.

A packet is considered a match if it matches any port name or number specified in the group. Only one port group may be specified. The port group must already be defined.

destination

Specifies a media access control (MAC) address to match. The address format is six 8-bit numbers, separated by colons, in hexadecimal; for example, 00:0a:59:9a:f2:ba.

Note: For policy-based routing, the usefulness of this parameter is limited because the MAC address is on a local interface.

Configuration mode

```
policy {
    route {
        pbr name {
                rule rule-number {
                    destination {
                        address address
                        mac-address address
                        port port
                    }
                }
            }
        }
    }
```

This match criterion specifies a group of addresses, ports, or networks for packet destination address.

A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the packet must be a match for both groups to be considered a match. For example, if both an address group and a port group are specified, the destination of the packet must match at least one item in the address group and at least one item in the port group.

An address group may be specified with a port group.

If both an address and a port are specified, the packet is considered a match only if both the address and the port match.

Use the `set` form of this command to create or modify a rule within an IP routing policy.

Use the `delete` form of this command to remove a rule from an IP routing policy.

Use the `show` form of this command to display a rule within an IP routing policy.

policy route pbr <name> rule <rule-number> disable

Disables a routing policy rule.

Syntax:

`set policy route pbr` *name* **`rule`** *rule-number* **`disable`**

Syntax:

`delete policy route pbr` *name* **`rule`** *rule-number* **`disable`**

Syntax:

`show policy route pbr` *name* **`rule`** *rule-number*

The rule is enabled.

name

The name of an IP routing policy.

rule-number

The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one `rule` configuration node.

Configuration mode

```
policy {
    route {
        pbr name {
                rule rule-number {
                    disable
                }
            }
        }
    }
```

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

Use the `set` form of this command to disable a routing policy rule.

Use the `delete` form of this command to re-enable a rule.

Use the `show` form of this command to display a routing policy rule.

policy route pbr <name> rule <rule-number> icmp <icmp>

Creates a routing policy rule to match Internet Control Message Protocol (ICMP) packets.

Syntax:

`set policy route pbr` *name* **`rule`** *rule-number* **`icmp`** { **`type`** *type-number* [**`code`** *code-number*] | **`name`** *name* }

Syntax:

`delete policy route pbr` *name* **`rule`** *rule-number* **`icmp`** [**`type`** [*number* **`code`**] | **`name`**]

Syntax:

`show policy route pbr` *name* **`rule`** *rule-number* **`icmp`** [**`type`** [*number* **`code`**] | **`name`**]

The rule is enabled.

name

Name of a PBR group. The PBR group must be unique and must not be used with other PBR policy commands.

rule-number

The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one `rule` configuration node.

icmp

The ICMP packet that matches the routing policy rule. The ICMP packet identifiers are type, code, and name.

type-number

An IPv4 ICMP type number. Values range from 0 through 255.

code-number

An IPv4 ICMP code number. Values range from 0 through 255.

name

Specifies matching for ICMP type names. The default name is **`any`**.

Configuration mode

```
policy {
    route {
        pbr name {
            rule rule-number {
                icmp {
                    type type-number {
                        code code-number
                    }
                    name name
                }
            }
        }
    }
}
```

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

> **Note:** As ICMP is an IPv4 protocol and ICMPv6 is an IPv6 protocol, configuring a routing policy rule to match ICMPv6 packets when address-family ipv4 is configured or vice versa are unlikely to be useful and probably will not behave as you are expecting it to behave.

You can specify an ICMP type code by type; for example, 128 (echo-request), or by a type and code pair; for example, type 1 and code 4 (port-unreachable). Alternatively, you can specify the ICMP type code explicitly by using the **`name`** *name* parameter ; for example, `name echo-request`.

For a list of ICMP codes and types, refer to ICMP Types *(page 47)*.

Use the `set` form of this command to create a rule to match ICMP packets.

Use the `delete` form of this command to delete a rule that matches ICMP packets.

Use the `show` form of this command to display a rule that matches ICMP packets.

policy route pbr <name> rule <rule-number> icmpv6 <icmpv6>

Creates a routing policy rule to match Internet Control Message Protocol (ICMP) IPv6 packets.

Syntax:
`set policy route pbr` *name* **`rule`** *rule-number* **`icmpv6`** { **`type`** *type-number* [**`code`** *code-number*] | **`name`** *name* }

Syntax:
`delete policy route pbr` *name* **`rule`** *rule-number* **`icmpv6`** [**`type`** [*number* **`code`**] | **`name`**]

Syntax:
`show policy route pbr` *name* **`rule`** *rule-number* **`icmpv6`** [**`type`** [*number* **`code`**] | **`name`**]

The rule is enabled.

name

Name of a PBR group. The PBR group must be unique and must not be used with other PBR policy commands.

rule-number

The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one `rule` configuration node.

icmpv6

The ICMPv6 packet that matches the routing policy rule. The ICMPv6 packet identifiers are type, code, and name.

type-number

An IPv6 ICMP type number. Values range from 0 through 255.

code-number

An IPv6 ICMP code number. Values range from 0 through 255.

name

Specifies matching for ICMPv6 type names. The default name is **any**.

Configuration mode

```
policy {
    route {
        pbr name {
            rule rule-number {
                icmpv6 {
                    type type-number {
                        code code-number
                    }
                    name name
                }
            }
        }
    }
}
```

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

> **Note:** As ICMP is an IPv4 protocol and ICMPv6 is an IPv6 protocol, configuring a routing policy rule to match ICMPv6 packets when address-family ipv4 is configured or vice versa are unlikely to be useful and probably will not behave as you are expecting it to behave.

You can specify an ICMPv6 type code by type; for example, 128 (echo-request), or by a type and code pair; for example, type 1 and code 4 (port-unreachable). Alternatively, you can specify the ICMPv6 type code explicitly by using the **`name`** *name* parameter; for example, `name echo-request`.

For a list of ICMPv6 codes and types, refer to ICMPv6 Types *(page 50)*.

Use the `set` form of this command to create a rule to match ICMPv6 packets.

Use the `delete` form of this command to delete a rule that matches ICMPv6 packets.

Use the `show` form of this command to view a rule that matches ICMPv6 packets.

policy route pbr <name> rule <rule-number> ipv6-route type <type-number>

Defines the IPv6 route type to match for a routing policy rule.

Syntax:

`set policy route pbr` *name* **`rule`** *rule-number* **`ipv6-route type`** *type-number*

Syntax:

`delete policy route pbr` *name* **`rule`** *rule-number* **`ipv6-route type`**

Syntax:

`show policy route pbr` *name* **`rule`** *rule-number* **`ipv6-route type`**

name

Name of a PBR group. The PBR group must be unique and must not be used with other PBR policy commands.

rule-number

The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one `rule` configuration node.

`ipv6-route`

Specifies matching based on an IPv6 route.

type-number

IPv6 route-type. Values range from 0 through 255.

Configuration mode

```
policy {
    route {
        pbr name {
            rule rule-number {
                ipv6-route {
                    type type-number
                }
            }
        }
    }
}
```

Note: This command can be used to block Type 0 routing headers in IPv6. RFC 5095 deprecates the use of Type 0 routing headers in IPv6 because they are a security risk.

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the`delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

Use the `set` form of this command to define the IPv6 route type for a routing-policy rule set.

Use the `delete` form of this command to delete the IPv6 route type for the routing-policy rule set.

Use the `show` form of this command to display the IPv6 route type for the routing-policy rule set.

policy route pbr <name> rule <rule-number> log

Enables logging for a routing policy rule.

Syntax:

`set policy route pbr` *name* **`rule`** *rule-number* **`log`**

Syntax:

`delete policy route pbr` *name* **`rule`** *number* **`log`**

Syntax:

`show policy route pbr` *name* **`rule`** *number*

Logging is disabled.

name

The name of an IP routing policy.

rule-number

The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one **`rule`** configuration node.

Configuration mode

```
policy {
    route {
        pbr name {
                rule rule-number {
                    log
                }
            }
        }
    }
```

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

When logging is enabled, any action taken is logged.

Use the `set` form of this command to enable logging for a routing policy rule.

Use the `delete` form of this command to restore the default behavior for logging, that is, actions are not logged.

Use the `show` form of this command to display whether logging is enabled or disabled.

policy route pbr <name> rule <rule-number> path-monitor monitor <monitor-name> policy <policy-name>

Defines a PBR policy rule for a Path Monitor and policy pair.

Syntax:
`set policy route pbr` *name* **`rule`** *rule-number* **`path-monitor monitor`** *monitor-name* **`policy`** *policy-name*

Syntax:
`delete policy route pbr` *name* **`rule`** *rule-number* **`path-monitor monitor`** *monitor-name* **`policy`** *policy-name*

Syntax:
`show policy route pbr` *name* **`rule`** *rule-number* **`path-monitor monitor`**

name
The name of an IP routing policy.
rule-number
The numeric identifier of an IP routing policy rule.
monitor-name
The name of a configured Path Monitor.
policy-name
The name of a configured Path Monitor policy.

Configuration mode

```
policy {
       route {
              pbr name {
                     rule rule-number {
                            path-monitor {
                                   monitor monitor-name {
                                          policy policy-name

                                   }
                            }
                     }
              }
       }
}
```

Use the `set` form of this command to define a PBR policy rule that identifies the traffic that matches the parameters defined by a Path Monitor and an associated policy. A match succeeds if the Path Monitor is Compliant or Marginally Compliant.

Use the `delete` form of this command to remove a Path Monitor or a Path Monitor policy from a PBR policy rule.

Use the `show` form of this command to display the Path Monitor and policy pairs that are configured for a PBR policy rule .

Note: A PBR rule treats a marginally compliant path as being compliant.

policy route pbr <name> rule <rule-number> port <port>

Defines the source port name, number, range, or port group for a routing policy rule.

Syntax:
`set policy route pbr` *name* **`rule`** *rule-number* { **`port`** [*port* | *1-65535* | *start-end* | *port-group-name*] }

Syntax:
`delete policy route pbr` *name* **`rule`** *rule-number* [**`port`** [*port* | *1-65535* | *start-end* | *port-group-name*]]

Syntax:
`show policy route pbr` *name* **`rule`** *number* [**`port`**]

name
The name of an IP routing policy.

rule-number
The numeric identifier of a policy rule. Rule numbers determine the order in which rules are processed. Each rule must have a unique rule number. The number ranges from 1 through 9999.

You can define multiple rules by creating more than one **`rule`** configuration node.

`port [ port | 1-65535 | start-end | port-group-name ]`
A source port to match. The format of the port is any of the following:

port-name: The name of an IP service; for example, http. You can specify any service name in the `/etc/services` file.

1-65535: A port number. The numbers range from 1 through 65535.

start-end: A specified range of ports; for example, 1001-1005.

port-group-name: A port group. A packet is considered a match if it matches any port name or number specified in the group. Only one port group may be specified. The port group must already be defined.

This criterion specifies a group of addresses, ports, or networks for packet source address.

A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the packet must be a match for both groups to be considered a match. For example, if both an address group and a port group are specified, the source of the packet must match at least one item in the address group and at least one item in the port group.

An address group may be specified with a port group.

If both an address and a port are specified, the packet is considered a match only if both the address and the port match.

Configuration mode

```
policy {
    route {
        pbr name {
                rule rule-number {
```

```
                    port name
                    port 1-65535
                    port start-end
                    port port-group-name
                }
            }
        }
    }
```

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

This criterion specifies a port or a group of ports for packet source address for a routing policy rule.

A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the packet must be a match for both groups in order to be considered a match. For example, if an address group and a port group are both specified, the packet's source must match at least one item in the address group and at least one item in the port group.

An address group can be specified together with a port group, and a network group can be specified together with a port group. You cannot specify both an address and a network group.

The address family must match the specified family by using the `set policy route pbr` *name* **`rule`** *number* **`address-family ipv4`** command.

Use the `set` form of this command to define the source for a routing policy rule.

Use the `delete` form of this command to remove the source for a routing policy rule.

Use the `show` form of this command to view the source for a routing policy rule.

policy route pbr <name> rule <rule-number> pcp <pcp-number>

Defines the 801.1 priority-code point number to match for a routing policy rule.

Syntax:
`set policy route pbr` *name* **`rule`** *rule-number* **`pcp`** *pcp-number*

Syntax:
`delete policy route pbr` *name* **`rule`** *rule-number* **`pcp`**

Syntax:
`show policy route pbr` *name* **`rule`** *rule-number* **`pcp`**

name
Name of a PBR group. The PBR group must be unique and must not be used with other PBR policy commands.

rule-number
The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one `rule` configuration node.

pcp-number
802.1 priority-code point number. Values range from 0 through 7.

Configuration mode

```
policy {
    route {
        pbr name {
            rule rule-number {
```

```
                pcp pcp-number
            }
        }
    }
}
```

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

Use the `set` form of this command to define an 802.1 priority-code point for a routing-policy rule set.

Use the `delete` form of this command to delete the 802.1 priority-code point for the routing-policy rule set.

Use the `show` form of this command to display the 802.1 priority-code point for the routing-policy rule set.

policy route pbr <name> rule <rule-number> protocol <protocol>

Defines the protocol of an IP routing policy rule.

Syntax:

`set policy route pbr` *name* **`rule`** *rule-number* **`protocol`** { *text* | *0-255* | **`all`** | *name* }

Syntax:

`delete policy route pbr` *name* **`rule`** *rule-number* **`protocol`** [*text* | *0-255* | **`all`** | *name*]

Syntax:

`show policy route pbr` *name* **`rule`** *rule-number* **`protocol`**

name

The name of an IP routing policy.

rule-number

The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one **`rule`** configuration node.

protocol

The *protocol* is any of the following:

text: Matches packets by protocol type. Any protocol literals or numbers listed in the file `/etc/protocols` can be specified. The keywords **`icmpv6`** and **`all`** (for all protocols) are also supported.

0-255: An IP protocol number that ranges from 0 through 255.

`all`: All IP protocols.

`!` *protocol*: All IP protocols except for the specified name or number. Prefixing the protocol name with the negation operator (the exclamation mark) matches every protocol except the specified protocol. For example, !tcp matches all protocols except TCP.

This parameter matches the last, next-header field in the IP header chain. This match means that if the packet has no extension headers, it matches the next-header field in the main header. If the packet does have extension headers, the parameter matches the next-header field of the last extension header in the chain. In other words, the parameter always matches the ID of the transport-layer packet that is being carried.

Exercise care when employing more than one rule that uses the negation. Routing policy rules are evaluated sequentially, and a sequence of negated rules could result in unexpected behavior.

Configuration mode

```
policy {
    route {
        pbr name {
```

```
                rule rule-number {
                    protocol
                        text
                        0-255
                        all
                        name
                }
            }
        }
    }
```

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

Note: The routing policy does not validate the protocol against the configured address-family. "protocol icmp" type is used with "address-family ipv4" while "protocol icmpv6" type is used with "address-family ipv6".

Use the `set` form of this command to define the protocol of an IP routing policy rule.

Use the `delete` form of this command to remove a protocol from a routing policy rule.

Use the `show` form of this command to view the protocol of a routing policy rule.

policy route pbr <name> rule <rule-number> source address <address>

Defines the source address for a routing policy rule.

Syntax:
`set policy route pbr` *name* **`rule`** *rule-number* **`source address`** *address*

Syntax:
`delete policy route pbr` *name* **`rule`** *rule-number* **`source address`** [*addresss*]

Syntax:
`show policy route pbr` *name* **`rule`** *rule-number* **`source`**

name
The name of an IP routing policy.

rule-number
The numeric identifier of a policy rule. Rule numbers determine the order in which rules are processed. Each rule must have a unique rule number. The number ranges from 1 through 9999.

You can define multiple rules by creating more than one **`rule`** configuration node.

`source`
Specifies matching based on a source address.

address
Specifies an address to match. Address formats are as follows:

address-group name: An address group that is configured with a list of addresses.

ip-address: An IPv4 address.

ip-address/prefix: An IPv4 network address, where 0.0.0.0/0 matches any network.

`!` *ip-address*: All IP addresses except the specified IPv4 address.

`!` *ip-address/prefix*: All IP addresses except the specified IPv4 network address.

ipv6-address: An IPv6 address; for example, fe80::20c:29fe:fe47:f89.

ip-address/prefix: An IPv6 network address, where ::/0 matches any network; for example, fe80::20c:29fe:fe47:f88/64.

- ! *ipv6-address*: All IP addresses except the specified IPv6 address.
- ! *ip-address/prefix*: All IP addresses except the specified IPv6 network address.

Configuration mode

```
policy {
    route {
        pbr name {
                rule rule-number {
                    source {
                        address address

                    }
                }
            }
        }
    }
```

This match criterion specifies a port or a group of ports for packet source address for a routing policy rule.

A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the packet must be a match for both groups to be considered a match. For example, if both an address group and a port group are specified, the source of the packet must match at least one item in the address group and at least one item in the port group.

An address group may be specified with a port group.

If both an address and a port are specified, the packet is considered a match only if both the address and the port match.

Use the `set` form of this command to define the source for a routing policy rule.

Use the `delete` form of this command to remove the source for a routing policy rule.

Use the `show` form of this command to view the source for a routing policy rule.

policy route pbr <name> rule <rule-number> source mac-address <address>

Defines the source MAC address to match for a routing policy rule.

Syntax:
`set policy route pbr` *name* **`rule`** *number* **`source mac-address`** *address*

Syntax:
`delete policy route pbr` *name* **`rule`** *number* **`source mac-address`** [*address*]

Syntax:
`show policy route pbr` *name* **`rule`** *number* **`source mac-address`** [*address*]

name
Name of a PBR group. The PBR group must be unique and must not be used with other PBR policy commands.

rule-number
The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one `rule` configuration node.

`source`
Specifies matching based on a source address.

address

Media access control (MAC) address. The address format is six 8-bit numbers, separated by colons, in hexadecimal; for example, 00:0a:59:9a:f2:ba.

Configuration mode

```
policy {
    route {
        pbr name {
            rule rule-number {
                source {
                    mac-address address
                }
            }
        }
    }
}
```

Note: For policy based routing, the usefulness of this command is limited because the MAC address is on a local interface.

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

Use the `set` form of this command to define a source MAC address for a routing-policy rule set.

Use the `delete` form of this command to delete the source MAC address for the routing-policy rule set.

Use the `show` form of this command to display the source MAC address for the routing-policy rule set.

policy route pbr <name> rule <rule-number> source port <port>

Defines the source port name, number, range, or port group for a routing policy rule.

Syntax:
`set policy route pbr` *name* **`rule`** *rule-number* **`source port`** [*name* | *1-65535* | *start-end* | *port-group-name*]

Syntax:
`delete policy route pbr` *name* **`rule`** *rule-number* **`source port`** [*name* | *1-65535* | *start-end* | *port-group-name*]

Syntax:
`show policy route pbr` *name* **`rule`** *rule-number* **`source port`**

name
The name of an IP routing policy.

rule-number
The numeric identifier of a policy rule. Rule numbers determine the order in which rules are processed. Each rule must have a unique rule number. The number ranges from 1 through 9999.

You can define multiple rules by creating more than one **`rule`** configuration node.

`source`
Specifies matching based on a source address.

`port [ name | 1-65535 | start-end | port-group-name ]`
A source port to match. The format of the port is any of the following:

name: The name of an IP service; for example, http. You can specify any service name in the `/etc/services` file.

1-65535: A port number. The numbers range from 1 through 65535.

start-end: A specified range of ports; for example, 1001-1005.

port-group-name: A port group. A packet is considered a match if it matches any port name or number specified in the group. Only one port group may be specified. The port group must already be defined.

This criterion specifies a group of addresses, ports, or networks for packet source address.

A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the packet must be a match for both groups to be considered a match. For example, if both an address group and a port group are specified, the source of the packet must match at least one item in the address group and at least one item in the port group.

An address group may be specified with a port group.

If both an address and a port are specified, the packet is considered a match only if both the address and the port match.

Configuration mode

```
policy {
    route {
        pbr name {
                rule rule-number {
                    source {
                        port name
                        port 1-65535
                        port start-end
                        port port-group-name
                    }
                }
            }
        }
    }
```

This criterion specifies a port or a group of ports for packet source address for a routing policy rule.

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the packet must be a match for both groups in order to be considered a match. For example, if an address group and a port group are both specified, the packet's source must match at least one item in the address group and at least one item in the port group.

Use the `set` form of this command to define the source for a routing policy rule.

Use the `delete` form of this command to remove the source for a routing policy rule.

Use the `show` form of this command to view the source for a routing policy rule.

policy route pbr <name> rule <rule-number> table <table-number>

Defines the table number for an IP routing policy rule.

Syntax:
`set policy route pbr` *name* **`rule`** *rule-number* **`table`** *table-number*

Syntax:
`delete policy route pbr` *name* **`rule`** *rule-number* **`table`** [*table-number*]

Syntax:

`show policy route pbr` *name* **`rule`** *rule-number*

name
The name of an IP routing policy. The policy name must be unique and must not be used with other PBR policy commands.

rule-number
The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one **`rule`** configuration node.

table-number
To match according to the PBR Table ID numbers 1 through 128. Performs alternate processing on packets satisfying the match criteria.

Configuration mode

```
policy {
    route {
        pbr name{
                rule rule-number {
                        table table-number
                }
            }
        }
    }
```

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **`rule`** *number* command to delete a routing policy.

Use the `set` form of this command to define the address family or routing table ID for an IP routing policy rule.

Use the `delete` form of this command to remove the address family or routing table ID for a rule.

Use the `show` form of this command to view the address family or routing table ID for a rule.

The address family must match the specified family by using the `set policy route pbr` *name* **`rule`** *number* **`address-family ipv4`** command.

Use the `set` form of this command to define the source for a routing policy rule.

Use the `delete` form of this command to remove the source for a routing policy rule.

Use the `show` form of this command to view the source for a routing policy rule.

policy route pbr <name> rule <rule-number> tcp flags <tcp-flag>

Defines the types of TCP flags to be matched for a routing policy rule.

Syntax:
`set policy route pbr` *name* **`rule`** *rule-number* **`tcp flags`** *flags*

Syntax:
`delete policy route pbr` *name* **`rule`** *rule-number* **`tcp flags`** [*flags*]

Syntax:
`show policy route pbr` *name* **`rule`** *rule-number* **`tcp flags`**

name
The name of an IP routing policy.

rule-number
The numeric identifier of the rule. Rule numbers determine the order in which rules are executed. Each rule must have a unique rule number. The numbers range from 1 through 9999.

You can define multiple rules by creating more than one **rule** configuration node.

tcp-flags

The flags to be matched in a packet. The flags are any of SYN, ACK, FIN, RST, URG, and PSH. You can specify more than one flag in a list separated by commas.

Prefixing a flag name with the negation operator matches packets with that flag unset. You can also use **!** to match packets by not using a given TCP flag. For example, the list SYN, **!**ACK, **!**FIN, **!**RST matches only packets with the SYN flag set and the ACK, FIN, and RST flags unset.

Configuration mode

```
policy {
    route {
        pbr name {
              rule rule-number {
                    tcp {
                       flags tcp-flags
                    }
                }
            }
        }
    }
```

You must specify the address-family, action, and table leaf nodes to configure a routing policy. It is recommended to use the `delete policy route pbr` *name* **rule** *number* command to delete a routing policy.

Use the `set` form of this command to define the types of TCP flags to be matched for a routing policy rule.

Use the `delete` form of this command to remove the types of TCP flags to be matched for a routing policy rule.

Use the `show` form of this command to view the types of TCP flags to be matched for a routing policy rule.

show application name <name>

Displays the type information for the DPI application.

Syntax:
`show application name` *name*

name

The name of a valid DPI application.

Operational mode

Use this command in operational mode to display the application name and the associated application types.

Use a valid application name.

For example:

The following example shows the application type information for the `zing` application.

Example: show application name

```
vyatta@vyatta:~$ show application name zing
'zing' (Zing.vn) is included in the following application types:
web classified_ads
vyatta@vyatta:~$
```

show application type <type>

Displays the application names associated with the given application type.

Syntax:
`show application type` *type*

type
The application type of a valid DPI application.

Operational mode

Use this command in operational mode to display the application names associated with the given application type.

Use a valid application type.

For example:

The following example shows the information for the DPI application type `email`.

Example: show application type

```
vyatta@vyatta:~$ show application type email
'email' includes the following applications:
lotus_live smtps smtp pop3s pop3 mapi lotusnotes linkedin imaps imap
vyatta@vyatta:~$
```

show policy route <interface>

Displays routing policy configuration or statistics.

Syntax:
`show policy route` *interface*

interface
The name of an interface.

Operational mode

A policy identifies traffic that matches parameters and specifies which table to use. The table defines the routes for a packet to take. A routing policy is a named collection of as many as 9,999 packet-classification rules. When applied to an interface, the policy rule classifies incoming traffic.

Note: The PBR rule counters count all of the matched packets regardless of the availability of the route.

Use this command in operational mode to display packet statistics for all PBR rules in all groups.

For example:

Example: show policy route

```
vyatta@vyatta:~$ show policy route
------------------------------------
Rulesets Information: PBR
------------------------------------
-------------------------------------------------------------------------------
PBR policy "myroute2":
Active on (dp0s3, in)
rule    action  proto           packets         bytes
----    ------  -----           -------         -----
10      drop    any             0               0
  condition - family inet apply dpi(youtube,none)
vyatta@vyatta:~$
```

show policy route table

Displays the configuration of the IP routing policy table.

Syntax:

```
show policy route table
```

Operational mode

Use this command to display the details about all the rules and tables configured for a IP routing policy.

The `show policy route table` command displays the following information:

```
vyatta@vyatta# show policy route table
PBR Group                    Rule  Table
---------------------------  ----  -----
                    myroute    10     1
                    myroute    20     2
                    myroute    10     1
                    myroute    20     2
```

Output field	Description
PBR Group	Name of a PBR group.
Rule	Number of the IP policy rule that is configured for a PBR group.
Table	Number of the PBR table that is configured for a PBR group.

Related commands

The following table lists related commands that are documented elsewhere.

Related commands documented elsewhere	
protocols static table	The commands for creating alternate routing tables are described in AT&T Vyatta Network Operating System Basic Routing Configuration Guide
resources group address-group <group-name>	Defines a group of IP addresses that are referenced in firewall rules. (Refer to AT&T Vyatta Network Operating System Basic Routing Configuration Guide.)
resources group port-group <group-name>	Defines a group of ports that are referenced in firewall rules. (Refer to AT&T Vyatta Network Operating System Basic Routing Configuration Guide.)
show ip route table	The command for displaying the contents of an alternate routing table is described in AT&T Vyatta Network Operating System Basic Routing Configuration Guide.

Related commands documented elsewhere	
firewall group	Routing policy match criteria support references to predefined groups of addresses, ports, and networks. Commands for defining such groups are described in AT&T Vyatta Network Operating System Firewall Configuration Guide.

ICMP Types

This appendix lists the Internet Control Messaging Protocol (ICMP) types defined by the Internet Assigned Numbers Authority (IANA).

The IANA has developed a standard that maps a set of integers onto ICMP types. The following table lists the ICMP types and codes defined by the IANA and maps them to the literal strings that are available in the AT&T Vyatta vRouter.

Table 6: ICMP types

ICMP Type	Code	Literal	Description
0 - Echo reply	0	echo-reply	Echo reply (pong)
3 - Destination unreachable		destination- unreachable	Destination is unreachable
	0	network-unreachable	Destination network is unreachable
	1	host-unreachable	Destination host is unreachable
	2	protocol-unreachable	Destination protocol is unreachable
	3	port-unreachable	Destination port is unreachable
	4	fragmentation-needed	Fragmentation is required
	5	source-route-failed	Source route has failed
	6	network-unknown	Destination network is unknown
	7	host-unknown	Destination host is unknown
	9	network-prohibited	Network is administratively prohibited
	10	host-prohibited	Host is administratively is prohibited
	11	ToS-network-unreachable	Network is unreachable for ToS
	12	ToS-host-unreachable	Host is unreachable for ToS

ICMP Type	Code	Literal	Description
	13	communication-prohibited	Communication is administratively prohibited
	14	host-precedence-violation	Requested precedence is not permitted.
	15	precedence-cutoff	Precedence is lower than the required minimum.
4 - Source quench	0	source-quench	Source is quenched (congestion control)
5 - Redirect message		redirect	Redirected message
	0	network-redirect	Datagram is redirected for the network
	1	host-redirect	Datagram is redirected for the host
	2	ToS-network-redirect	Datagram is redirected for the ToS and network
	3	ToS-host-redirect	Datagram is redirected for the ToS and host
8 - Echo request	0	echo-request	Echo request (ping)
9 - Router advertisement	0	router-advertisement	Router advertisement
10 - Router solicitation	0	router-solicitation	Router solicitation
11 - Time exceeded		time-exceeded	Time to live (TTL) has exceeded
	0	ttl-zero-during-transit	TTL has expired in transit
	1	ttl-zero-during-reassembly	Fragment reassembly time has exceeded
12 - Parameter problem: Bad IP header		parameter-problem	Bad IP header
	0	ip-header-bad	Pointer that indicates an error
	1	required-option-missing	Missing required option
13 - Timestamp	0	timestamp-request	Request for a timestamp
14 - Timestamp reply	0	timestamp-reply	Reply to a request for a timestamp

ICMP Type	Code	Literal	Description
15 - Information request	0		Information request
16 - Information reply	0		Information reply
17 - Address mask request	0	address-mask-request	Address mask request
18 - Address mask reply	0	address-mask-reply	Address mask reply

ICMPv6 Types

This appendix lists the ICMPv6 types defined by the Internet Assigned Numbers Authority (IANA).

The Internet Assigned Numbers Authority (IANA) has developed a standard that maps a set of integers onto ICMPv6 types. The following table lists the ICMPv6 types and codes defined by the IANA and maps them to the strings literal strings available in the AT&T Vyatta vRouter.

Table 7: ICMPv6 types

ICMPv6 Type	Code	Literal	Description
1 - Destination unreachable		destination- unreachable	
	0	no-route	No route to destination
	1	communication-prohibited	Communication with destination administratively prohibited
	2		Beyond scope of source address
	3	address-unreachable	Address unreachable
	4	port-unreachable	Port unreachable
	5		Source address failed ingress/egress policy
	6		Reject route to destination
2 - Packet too big	0	packet-too-big	
3 - Time exceeded		time-exceeded	
	0	ttl-zero-during-transit	Hop limit exceeded in transit
	1	ttl-zero-during-reassembly	Fragment reassembly time exceeded
4 - Parameter problem		parameter-problem	
	0	bad-header	Erroneous header field encountered
	1	unknown-header-type	Unrecognized Next Header type encountered

ICMPv6 Type	Code	Literal	Description
	2	unknown-option	Unrecognized IPv6 option encountered
128 - Echo request	0	echo-request	Echo request (ping)
129 - Echo reply	0	echo-reply	Echo reply (pong)
133 - Router solicitation	0	router-solicitation	Router solicitation
134 - Router advertisement	0	router-advertisement	Router advertisement
135 - Neighbor solicitation	0	neighbor-solicitation (neighbour-solicitation)	Neighbor solicitation
136 - Neighbor advertisement	0	neighbor-advertisement (neighbour-advertisement)	Neighbor advertisement

Supported Interface Types

The following table shows the syntax and parameters of supported interface types. Depending on the command, some of these types may not apply.

Interface Type	Syntax	Parameters
Bridge	`bridge` *brx*	*brx*: The name of a bridge group. The name ranges from br0 through br999.

Interface Type	Syntax	Parameters
Data plane	`dataplane` *interface-name*	*interface-name*: The name of a data plane interface. Following are the supported formats of the interface name: • `dp`*x*`p`*y*`p`*z*—The name of a data plane interface, where — `dp`*x* specifies the data plane identifier (ID). Currently, only dp0 is supported. — `p`*y* specifies a physical or virtual PCI slot index (for example, p129). — `p`*z* specifies a port index (for example, p1). For example, dp0p1p2, dp0p160p1, and dp0p192p1. • `dp`*x*`em`*y* —The name of a data plane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where `em`*y* specifies an embedded network interface number (typically, a small number). For example, dp0em3. • `dp`*x*`s`*y*—The name of a data plane interface in a system in which the BIOS identifies the network interface card to reside in a particular physical or virtual slot *y*, where *y* is typically a small number. For example, for the dp0s2 interface, the BIOS identifies slot 2 in the system to contain this interface. • `dp`*x*`P`*n*`p`*y*`p`*z* —The name of a data plane interface on a device that is installed on a secondary PCI bus, where `P`*n* specifies the bus number. You can use this format to name data plane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of *n* must be an integer greater than 0. For example, dp0P1p162p1 and dp0P2p162p1.

Interface Type	Syntax	Parameters
Data plane vif	`dataplane` *interface-name* `vif` *vif-id* [`vlan` *vlan-id*]	*interface-name*: Refer to the preceding description. *vif-id*: A virtual interface ID. The ID ranges from 1 through 4094. *vlan-id*: The VLAN ID of a virtual interface. The ID ranges from 1 through 4094.
Loopback	`loopback lo` or `loopback lo`*n*	*n*: The name of a loopback interface, where *n* ranges from 1 through 99999.
OpenVPN	`openvpn` *vtunx*	*vtunx*: The identifier of an OpenVPN interface. The identifier ranges from vtun0 through vtun*x*, where *x* is a nonnegative integer.
Tunnel	`tunnel` *tunx* or `tunnel` *tunx* `parameters`	*tunx*: The identifier of a tunnel interface you are defining. The identifier ranges from tun0 through tun*x*, where *x* is a nonnegative integer.
Virtual tunnel	`vti` *vtix*	*vtix*: The identifier of a virtual tunnel interface you are defining. The identifier ranges from vti0 through vti*x*, where *x* is a nonnegative integer. **Note:** Before you can configure a vti interface, you must configure a corresponding vpn. **Note:** This interface does not support IPv6.
VRRP	*parent-interface* `vrrp vrrp-group` *group*	*parent-interface*: The type and identifier of a parent interface; for example, data plane dp0p1p2 or bridge br999. *group*: A VRRP group identifier. The name of a VRRP interface is not specified. The system internally constructs the interface name from the parent interface identifier plus the VRRP group number; for example, dp0p1p2v99. Note that VRRP interfaces support the same feature set as does the parent interface.

VRF support for PBR

The implementation of VRF on the AT&T Vyatta vRouter supports policy-based routing (PBR).

Command support for VRF routing instances

VRF allows an AT&T Vyatta vRouter to support multiple routing tables, one for each VRF routing instance. Some commands in this guide support VRF and can be applied to particular routing instances.

Use the guidelines in this section to determine correct syntax when adding VRF routing instances to commands. For more information about VRF, refer to AT&T Vyatta Network Operating System Basic Routing Configuration Guide. This guide includes an overview of VRF, VRF configuration examples, information about VRF-specific features, and a list of commands that support VRF routing instances.

Adding a VRF routing instance to a Configuration mode command

For most Configuration mode commands, specify the VRF routing instance at the beginning of a command. Add the appropriate VRF keywords and variable to follow the initial action (**`set`**, **`show`**, or **`delete`**) and before the other keywords and variables in the command.

Example: Configuration mode example: syslog

The following command configures the syslog logging level for the specified syslog host. The command does not include a VRF routing instance, so the command applies to the default routing instance.

```
vyatta@R1# set system syslog host 10.10.10.1 facility all level debug
vyatta@R1# show system syslog
syslog {
    host 10.10.10.1 {
            facility all {
                    level debug
            }
    }
}
```

The following example shows the same command with the VRF routing instance (GREEN) added. Notice that **`routing routing-instance GREEN`** has been inserted between the basic action (**set** in the example) and the rest of the command. Most Configuration mode commands follow this convention.

```
vyatta@R1# set routing routing-instance GREEN system syslog host 10.10.10.1 facility all
 level debug
vyatta@R1# show routing
routing {
     routing-instance GREEN {
             system {
                     syslog {
                             host 11.12.13.2:514 {
                                     facility all {
                                             level debug
                                     }
                             }
                     }
             }
     }
}
```

Example: Configuration mode example: SNMP

Some features, such as SNMP, are not available on a per-routing instance basis but can be bound to a specific routing instance. For these features, the command syntax is an exception to the convention of specifying the routing instance at the beginning of Configuration mode commands.

The following example shows how to configure the SNMPv1 or SNMPv2c community and context for the RED and BLUE routing instances. The first two commands specify the RED routing instance as the context for community A and BLUE routing instance as the context for community B. The subsequent commands complete the configuration.

For more information about configuring SNMP, refer to AT&T Vyatta Network Operating System Remote Management Configuration Guide.

```
vyatta@R1# set service snmp community commA context RED
vyatta@R1# set service snmp community commB context BLUE
vyatta@R1# set service snmp view all oid 1
vyatta@R1# set service snmp community commA view all
vyatta@R1# set service snmp community commB view all
vyatta@R1# show service snmp community
 community commA {
        context RED
        view all
 }
 community commB {
        context BLUE
        view all
 }
[edit]
vyatta@vyatta#
```

Adding a VRF routing instance to an Operational mode command

The syntax for adding a VRF routing instance to an Operational mode command varies according to the type of command parameters:

- If the command does not have optional parameters, specify the routing instance at the end of the command.
- If the command has optional parameters, specify the routing instance after the required parameters and before the optional parameters.

Example: Operational mode examples without optional parameters

The following command displays dynamic DNS information for the default routing instance.

```
vyatta@vyatta:~$ show dns dynamic status
```

The following command displays the same information for the specified routing instance (GREEN). The command does not have any optional parameters, so the routing instance is specified at the end of the command.

```
vyatta@vyatta:~$ show dns dynamic status routing-instance GREEN
```

Example: Operational mode example with optional parameters

The following command obtains multicast path information for the specified host (10.33.2.5). A routing instance is not specified, so the command applies to the default routing instance.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 detail
```

The following command obtains multicast path information for the specified host (10.33.2.5) and routing instance (GREEN). Notice that the routing instance is specified before the optional **detail** keyword.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 routing-instance GREEN detail
```

Example: Operational mode example output: SNMP

The following SNMP **show** commands display output for routing instances.

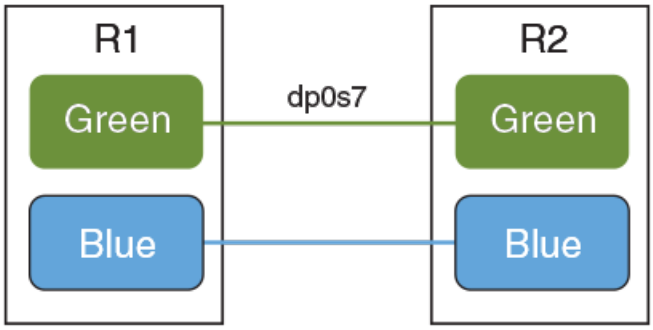
```
vyatta@vyatta:~$ show snmp routing-instance
Routing Instance SNMP Agent is Listening on for Incoming Requests:
Routing-Instance            RDID
-----------------           ----
RED                         5

vyatta@vyatta:~$ show snmp community-mapping
SNMPv1/v2c Community/Context Mapping:
Community                   Context
---------                   -------
commA                       'RED'
commB                       'BLUE'
deva                        'default'


vyatta@vyatta:~$ show snmp trap-target
SNMPv1/v2c Trap-targets:
Trap-target                   Port   Routing-Instance Community
-----------                   ----   ---------------- ---------
1.1.1.1                              'RED'           'test'


vyatta@vyatta:~$ show snmp v3 trap-target
SNMPv3 Trap-targets:
Trap-target                   Port   Protocol Auth Priv Type   EngineID              Routing-
Instance User
-----------                   ----   -------- ---- ---- ----   --------
 ---------------- ----
2.2.2.2                       '162'  'udp'    'md5'      'infor                        'BLUE'
         'test'
```

Configuring policy-based routing on a routing instance

In this example, the R1 vRouter is connected to the R2 vRouter through the dp0s7 interface that is bound to the GREEN routing instance.

The following steps show how to create an alternate routing table in the GREEN routing instance on dp0s7.

Figure 2: Configuring policy-based routing on a routing instance



To configure policy-based routing on a vRouter perform the following configuration and then reproduce the configuration as described in AT&T Vyatta Network Operating System Basic Routing Configuration Guide.

Table 8: Configuring policy-based static routes on a routing instance

Step	Command
Define the dp0s7 interface and bind it to the GREEN routing instance.	`vyatta@R1# set interfaces dataplane dp0s7` `vyatta@R1# set routing routing-instance GREEN interface dp0s7`
Define an interface based static route.	`vyatta@R1# set routing routing-instance GREEN protocols static table 10 interface-route 20.1.1.0/24 next-hop-interface dp0s7` `vyatta@R1# set routing routing-instance GREEN protocols static table 10 interface-route6 2010::/64 next-hop-interface dp0s7`

Create IPv4 and IPv6 PBR static routes under the GREEN routing instance.	`vyatta@R1# set routing routing-instance GREEN protocols static table 10 route 20.1.1.0/24 next-hop 10.1.1.2 interface dp0s7`
	
	`vyatta@R1# set routing routing-instance GREEN protocols static table 10 route6 2010::/64 next-hop 1010::2 interface dp0s7`

Create the IPv4 and IPv6 static routes with distance in the GREEN routing instance in the 10 PBR table.	`vyatta@R1# set routing routing-instance GREEN protocols static table 10 route 20.1.1.0/24 next-hop 10.1.1.2 distance 8` `vyatta@R1# set routing routing-instance GREEN protocols static table 10 route6 2010::/64 next-hop 1010::2 distance 8`
Create IPv4 and IPv6 black hole PBR static route configurations under the GREEN routing instance.	`vyatta@R1# set routing routing-instance GREEN protocols static table 10 route 20.1.1.0/24 blackhole` `vyatta@R1# set routing routing-instance GREEN protocols static table 10 route6 2010::/64 blackhole`

Step	Command
Create unreachable IPv4 and IPv6 PBR static routes under the GREEN routing instance.	`vyatta@R1# set routing routing-instance GREEN protocols static route table 10 route 20.1.1.0/24 unreachable` `vyatta@R1# set routing routing-instance GREEN protocols static route6 table 10 route6 2010::/64 unreachable`

View the configuration.	`vyatta@R1# show routing`
	`routing {`
	`  routing-instance GREEN {`
	`  interface dp0s7`
	`  protocols {`
	`  static {`
	`  table 10 {`
	`  interface-route 20.1.1.0/24 {`
	`  next-hop-interface dp0s7`
	`  }`
	`  interface-route6 2010::/64 {`
	`  next-hop-interface dp0s7`
	`  }`
	`  route 20.1.1.0/24 {`
	`  blackhole`
	`  next-hop 10.1.1.2 {`
	`  distance 10`
	`  interface dp0s7`
	`  }`
	`  unreachable`
	`  }`
	`  route6 2010::/64 {`
	`  blackhole`
	`  next-hop 1010::2 {`
	`  distance 10`
	`  interface dp0s5`
	`  }`
	`  unreachable`
	`  }`
	`  }`
	`  }`
	`  }`
	`}`

List of Acronyms

Acronym	Description
ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMVPN	dynamic multipoint VPN
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers

Acronym	Description
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM

Acronym	Description
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access