

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Configure](#)

[Network Diagram](#)

[Configurations](#)

[Verify](#)

[Case 1:Parent route is learnt via a static route on border routers](#)

[Case 2:Parent route is learnt via OSPF](#)

[Related Cisco Support Community Discussions](#)

Introduction

This document describes how PfRv2 (Performance Routing) controls traffic based on PfRv2 policy decision. This document discusses use of static routes and policy based routing in PfRv2.

Prerequisites

Requirements

Cisco recommends that you have basic knowledge of Performance Routing (PfR).

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Configure

PfRv2 allows a network administrator to configure policies and accordingly route the traffic as per PfRv2 policy outcome. There are various modes in which PfRv2 controls traffic and it depends on the protocol via which the parent route for destination prefix is learnt. PfRv2 is capable of changing routing information base (RIB) by manipulating routing protocols, injecting static routes or via dynamic policy based routing.

- If the parent route is learnt via BGP, PfRv2 can dynamically manipulate routes using attributes like local preference.
- If the parent route is learnt via EIGRP, PfRv2 can inject a new route in EIGRP topology table.
- If the parent route is learnt via static route, PfR2 injects a more specific(better) route on PfR selected border router (BR).
- If parent route is learnt via none of the above three mechanisms, PfRv2 uses policy based

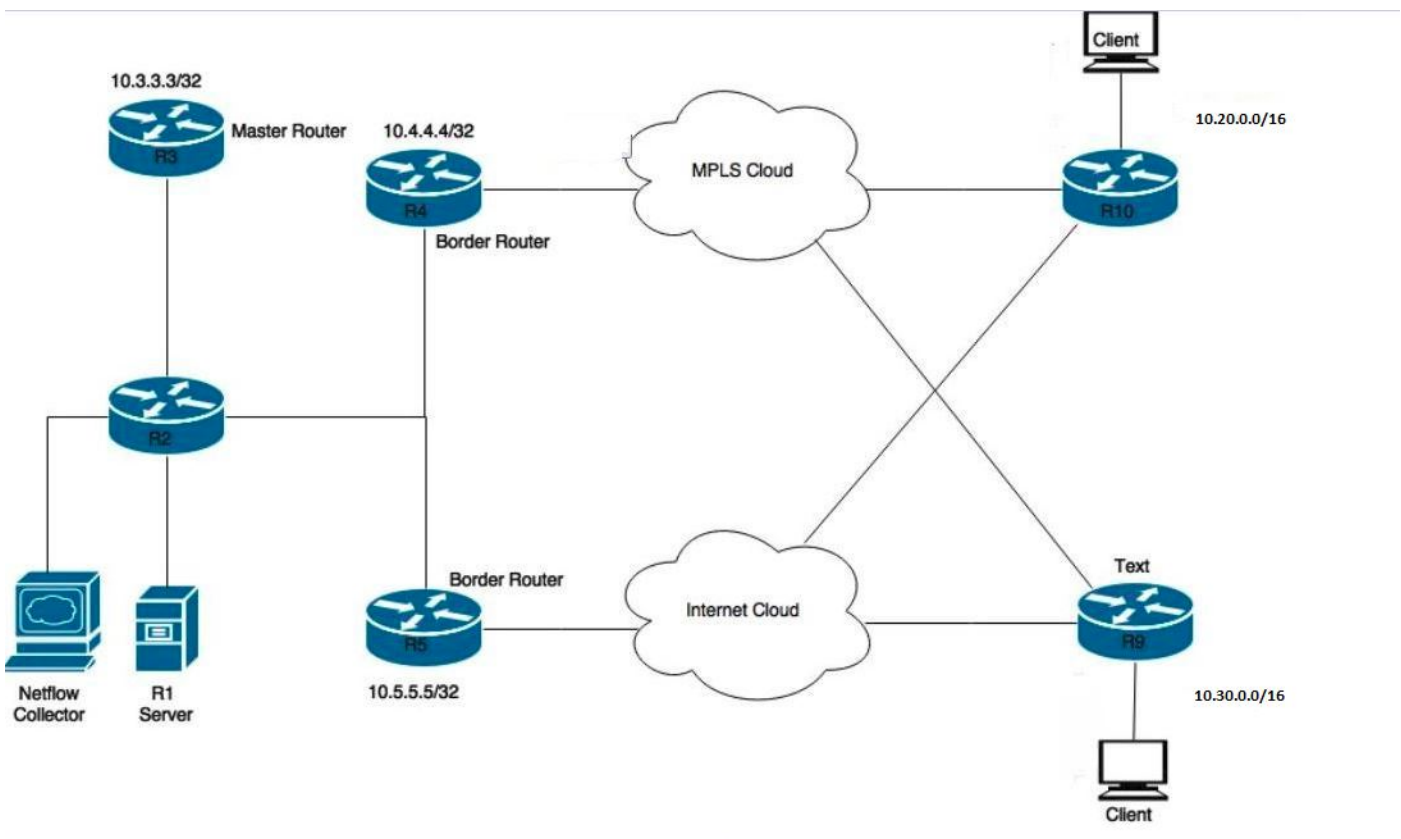
routing (PBR) to push traffic over selected BR.

Parent Route	Prefix control method
BGP	BGP
EIGRP	EIGRP
Static route	Static route
OSPF,ISIS,RIP etc	PBR

This article discusses PfRv2 using static routes (when parent route is via static route) and PBR (when parent route in RIB is via RIP,OSPF,ISIS etc) to control traffic.

Network Diagram

This document would refer following image as a sample topolgy for rest of the document.



Devices shown in the diagram:

R1- Server, Initiating traffic.

R3- PfR Master Router.

R4 & R5- PfR Border Router.

Clients connected to R9 & R10 are devices receiving the traffic from the R1 server.

Configurations

In this scenario two learn lists will be configured, one for application (APPLICATION-LEARN-LIST) and data (DATA-LEARN-LIST) traffic. This scenario uses a prefix-list to define traffic. An access-list can also be used to match traffic type like TCP, UDP, ICMP etc. DSCP and TOS can also be used to define your traffic.

Verify

Case 1: Parent route is learnt via a static route on border routers

In this scenario, traffic is flowing for destinations 10.20.20.1 and 10.30.30.1. Below is how parent route looks like on R4 and R5.

R4#show ip route

```
--output suppressed--
S      10.20.0.0/16 [1/0] via 10.0.68.8
S      10.30.0.0/16 [1/0] via 10.0.68.8
```

R5#show ip route

```
--output suppressed--
S      10.20.0.0/16 [1/0] via 10.0.57.7
S      10.30.0.0/16 [1/0] via 10.0.57.7
```

When traffic flows, PfRv2 learns traffic prefixes and traffic falls into INPOLICY state as shown below in output.

R3#show pfr master traffic-class

OER Prefix Statistics:

```
--output suppressed--
```

DstPrefix	Appl_ID	Dscp	Prot	SrcPort	DstPort	SrcPrefix	Flags	State	Time	CurrBR	CurrI/F	Protocol
	PasSDly	PasLDly	PasSun	PasLUn	PasSLos	PasLLos	EBw	IBw				
	ActSDly	ActLDly	ActSun	ActLUn	ActSJit	ActPMOS	ActSLos	ActLLos				
10.20.20.0/24			N	N	N	N	N	N				
			INPOLICY	31		10.4.4.4	Et1/0					STATIC
	N	N	N	N	N	N	N	N				
	1	2	0	0	N	N	N	N				
10.30.30.0/24			N	N	N	N	N	N				
			INPOLICY	30		10.5.5.5	Et1/0					STATIC
	N	N	N	N	N	N	N	N				
	4	2	0	0	N	N	N	N				

As can be seen below that R4 (10.4.4.4) router injected a more specific route 10.20.20.0/24. This auto generated route is automatically tagged with a tag value of 5000. This more specific better route makes R4 as better BR for traffic leaving for 10.20.20.0/24.

R4#show pfr border routes static

Flags: C - Controlled by oer, X - Path is excluded from control,
E - The control is exact, N - The control is non-exact

Flags	Network	Parent	Tag
CE	10.20.20.0/24	10.20.0.0/16	5000
XN	10.30.30.0/24		

```
R4#show ip route 10.20.20.0 255.255.255.0
```

```
Routing entry for 10.20.20.0/24
```

```
Known via "static", distance 1, metric 0
```

```
Tag 5000
```

```
Redistributing via ospf 100
```

```
Routing Descriptor Blocks:
```

```
* 10.0.46.6, via Ethernet1/0
```

```
Route metric is 0, traffic share count is 1
```

```
Route tag 5000
```

Likewise similar behavior can be seen on R5 and it injects a more specific route 10.30.30.0/24 as well which has a tag of 5000. This makes R5 a suitable candidate to route traffic for 10.30.30.0/24. This is how PfRv2 prefer traffic to be routed as shown above in "show pfr master traffic-class".

```
R5#show pfr border routes static
```

```
Flags: C - Controlled by oer, X - Path is excluded from control,  
       E - The control is exact, N - The control is non-exact  
Flags Network          Parent          Tag  
XN   10.20.20.0/24  
CE   10.30.30.0/24     10.30.0.0/16     5000
```

```
R5#show ip route 10.30.30.0 255.255.255.0  
Routing entry for 10.30.30.0/24  
  Known via "static", distance 1, metric 0  
  Tag 5000  
  Redistributing via ospf 100  
  Routing Descriptor Blocks:  
 * 10.0.57.7, via Ethernet1/0  
   Route metric is 0, traffic share count is 1  
   Route tag 5000
```

In the event there are multiple border routers (like in this case), these auto generated static routes have to be manually redistributed into IGP so as it could reach other border routers and they could route traffic based on the more specific route generated by selected BR.

Case 2: Parent route is learnt via OSPF

Any parent route which is not learnt via BGP, EIGRP or static route is controlled using policy based routing (PBR). PFRv2 injects dynamic route-map and access-list to control traffic. Below is how OSPF parent route looks like on R4 and R5.

```
R4#show ip route  
--output suppressed--  
O E2    10.20.0.0/16 [110/20] via 10.0.46.6, 02:16:35, Ethernet1/0  
O E2    10.30.0.0/16 [110/20] via 10.0.46.6, 02:16:35, Ethernet1/0
```

```
R5#show ip route  
--output suppressed--  
O E2    10.20.0.0/16 [110/20] via 10.0.57.7, 02:18:20, Ethernet1/0  
O E2    10.30.0.0/16 [110/20] via 10.0.57.7, 02:18:20, Ethernet1/0
```

When PFRv2 has to manipulate traffic flow via policy based routing, it requires a directly connected interface between BRs. This directly connected link could be a physical connection or it could be a GRE tunnel. This tunnel has to be manually created and configured as internal interface in PFRv2 border definition.

```
R4#show ip route  
--output suppressed--  
O E2    10.20.0.0/16 [110/20] via 10.0.46.6, 02:16:35, Ethernet1/0  
O E2    10.30.0.0/16 [110/20] via 10.0.46.6, 02:16:35, Ethernet1/0
```

```
R5#show ip route  
--output suppressed--  
O E2    10.20.0.0/16 [110/20] via 10.0.57.7, 02:18:20, Ethernet1/0  
O E2    10.30.0.0/16 [110/20] via 10.0.57.7, 02:18:20, Ethernet1/0
```

As per PFRv2 defined policy, it comes out with best exit router (BR) for 10.20.20.0/24 and 10.30.30.0/24. For example in the event when traffic destined for 10.20.20.0/24 comes to R5 (10.5.5.5) which is not the selected BR, a dynamic route-map and access-list is automatically injected to policy route the traffic to selected BR R4 (10.4.4.4). Packets are policy routed over the tunnel interface that was defined earlier.

```
R4#show ip route  
--output suppressed--
```

```
O E2      10.20.0.0/16 [110/20] via 10.0.46.6, 02:16:35, Ethernet1/0
O E2      10.30.0.0/16 [110/20] via 10.0.46.6, 02:16:35, Ethernet1/0
```

R5#show ip route

--output suppressed--

```
O E2      10.20.0.0/16 [110/20] via 10.0.57.7, 02:18:20, Ethernet1/0
O E2      10.30.0.0/16 [110/20] via 10.0.57.7, 02:18:20, Ethernet1/0
```