

Restrição de destino de rota

Contents

[Introduction](#)

[Finalidade da restrição de destino de rota](#)

[Comportamento sem RTC](#)

[Configuração RTC](#)

[Configuração do PE](#)

[Configuração RR](#)

[Comportamento do RTC](#)

[PE](#)

[RR](#)

[Tratamento de Atualização de Rotas](#)

[Informações Relacionadas](#)

Introduction

Este documento descreve um mecanismo pelo qual a troca de prefixos VPNv4 e VPNv6 em direção aos roteadores Provider Edge (PE) é reduzida ao mínimo necessário.

Finalidade da restrição de destino de rota

Com a VPN MPLS (Multiprotocol Label Switching), o peer iBGP (Border Gateway Protocol) ou RR (Route Reflector) envia todos os prefixos VPN4 e/ou VPN6 aos roteadores PE. O roteador PE descarta os prefixos VPN4/6 para os quais não há importação de roteamento e encaminhamento de VPN (VRF). Esse é um comportamento em que o RR envia prefixos VPN4/6 para o roteador PE, que ele não precisa. Isso é um desperdício de poder de processamento no RR e no PE e um desperdício de largura de banda.

Com RTC (Route Target Constraint, Restrição de Destino de Rota), o RR envia somente prefixos VPN4/6 desejados para o PE. 'Desejado' significa que o PE tem VRF importando os prefixos específicos.

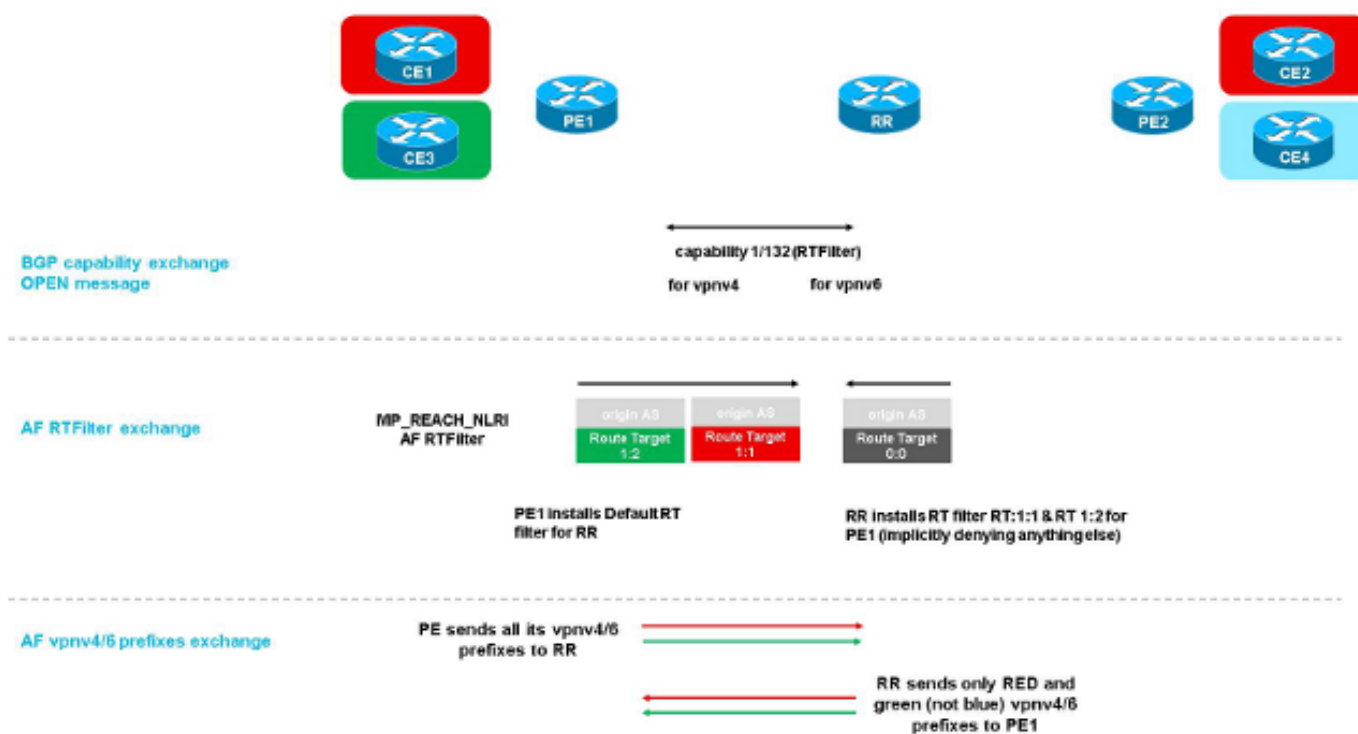
RFC 4684 especifica RTC. O suporte é por meio de um novo filtro de família de endereços para VPNv4 e VPNv6.

As informações de filtragem de RT (Route Target) são obtidas da lista de importação de RT de VPN de todos os VRFs do roteador PE. O roteador PE envia essas informações de filtragem como uma atualização BGP no filtro da família de endereços para o RR. Essas informações de filtragem ou associação RT são codificadas nos atributos Network Layer Reachability Information (NLRI) dos atributos MP_REACH_NLRI e MP_UNREACH_NLRI.

O peer BGP receptor converte esse NLRI em um filtro e instala esse filtro de saída para o peer emissor. O peer BGP receptor usa esse filtro para decidir quais prefixos VPNv4/6 enviar ou não, dependendo da presença de RTs conectadas.

Para que o RTC funcione, ambos os peers BGP precisam suportar o RTC. Ou seja, o RR e o PE precisam de o apoiar. No entanto, a implantação pode ser incremental, o que significa que nem todos os roteadores RR e PE precisam suportá-la de uma só vez. O RTC pode funcionar na rede, com alguns roteadores PE suportando-o e outros não. Nos roteadores que o suportam, o RTC já estará ativo. Nos roteadores que ainda não oferecem suporte a ele, os anúncios funcionarão como antes, o que é sem RTC (portanto sem qualquer filtragem de saída).

Esta figura mostra o princípio do RTC:



Comportamento sem RTC

O RR envia todos os prefixos VPN4/6 ao PE. O PE descarta aqueles para os quais não há importação da RT. As atualizações do BGP de depuração mostram os prefixos descartados. A mensagem 'NEGADO devido a: extended community not supported' (comunidade estendida sem suporte) é fornecida.

Um exemplo para unicast VPNv4 é o seguinte:

```
BGP(4): 10.100.1.3 rcvd UPDATE w/ att: nexthop 10.100.1.1, origin i, localpref 100,
metric 0, originator 10.100.1.1, clusterlist 10.100.1.3, merged path 65003,
AS_PATH , extended community RT:1:2
BGP(4): 10.100.1.3 rcvd 1:2:10.100.1.6/32, label 27 -- DENIED due to: extended
community not supported;
```

Um exemplo para unicast VPNv6 é o seguinte:

```
BGP(5): 10.100.1.3 rcvd UPDATE w/ attr: nexthop ::FFFF:10.100.1.1, origin i,  
localpref 100, metric 0, originator 10.100.1.1, clusterlist 10.100.1.3,  
merged path 65003, AS_PATH , extended community RT:1:2  
BGP(5): 10.100.1.3 rcvd [1:2]2001:10:100:1::6/128, label 23 -- DENIED due to:  
extended community not supported;
```

Configuração RTC

Configuração do PE

```
vrf definition green  
rd 1:2  
route-target export 1:2  
route-target import 1:2  
!  
address-family ipv4  
exit-address-family  
!  
vrf definition red  
rd 1:1  
route-target export 1:1  
route-target import 1:1  
!  
address-family ipv4  
exit-address-family  
!  
address-family ipv6  
exit-address-family  
  
router bgp 1  
bgp log-neighbor-changes  
neighbor 10.100.1.3 remote-as 1  
neighbor 10.100.1.3 update-source Loopback0  
neighbor 10.100.1.4 remote-as 1  
neighbor 10.100.1.4 update-source Loopback0  
!  
address-family vpnv4  
neighbor 10.100.1.3 activate  
neighbor 10.100.1.3 send-community both  
neighbor 10.100.1.4 activate  
neighbor 10.100.1.4 send-community both  
exit-address-family  
!  
address-family rtfiler unicast  
neighbor 10.100.1.3 activate  
neighbor 10.100.1.3 send-community extended  
exit-address-family  
!  
address-family ipv4 vrf green  
neighbor 10.1.6.6 remote-as 65003  
neighbor 10.1.6.6 activate  
neighbor 10.1.6.6 send-community both  
exit-address-family  
!  
address-family ipv4 vrf red
```

```
neighbor 10.1.5.5 remote-as 65001
neighbor 10.1.5.5 activate
neighbor 10.1.5.5 send-community both
exit-address-family
```

Configuração RR

```
router bgp 1
  bgp log-neighbor-changes
  neighbor 10.100.1.1 remote-as 1
  neighbor 10.100.1.1 update-source Loopback0
  neighbor 10.100.1.2 remote-as 1
  neighbor 10.100.1.2 update-source Loopback0
  !
  address-family vpnv4
  neighbor 10.100.1.1 activate
  neighbor 10.100.1.1 send-community both
  neighbor 10.100.1.1 route-reflector-client
  neighbor 10.100.1.2 activate
  neighbor 10.100.1.2 send-community both
  neighbor 10.100.1.2 route-reflector-client
  exit-address-family
  !
  address-family rtfiler unicast
  neighbor 10.100.1.1 activate
  neighbor 10.100.1.1 send-community both
  neighbor 10.100.1.1 route-reflector-client
  neighbor 10.100.1.1 default-originate
  exit-address-family
```

Comportamento do RTC

Quando o peering do BGP é estabelecido, os peers trocam a capacidade de filtro, que é 1/132 (para VPNV4 e VPNV6).

```
RR1# show bgp rtfiler unicast all neighbors 10.100.1.1
BGP neighbor is 10.100.1.1, remote AS 1, internal link
  BGP version 4, remote router ID 10.100.1.1
  BGP state = Established, up for 00:14:28
  Last read 00:00:01, last write 00:00:56, hold time is 180,
  keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: received
    Address family VPNv4 Unicast: advertised and received
    Address family VPNv6 Unicast: advertised and received
    Address family RT Filter: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
      Stateful switchover support enabled: NO for session 1
  Message statistics:
    InQ depth is 0
```

OutQ depth is 0

	Sent	Rcvd
Opens:	1	1
Notifications:	0	0
Updates:	6	7
Keepalives:	17	18
Route Refresh:	0	0
Total:	24	30

Default minimum time between advertisement runs is 0 seconds

For address family: VPNv4 Unicast

Session: 10.100.1.1

BGP table version 65, neighbor version 65/0

Output queue size : 0

Index 19, Advertise bit 1

Route-Reflector Client

19 update-group member

RT Filter activate

Community attribute sent to this neighbor

Slow-peer detection is disabled

Slow-peer split-update-group dynamic is disabled

	Sent	Rcvd
--	------	------

...

For address family: VPNv6 Unicast

Session: 10.100.1.1

BGP table version 5, neighbor version 5/0

Output queue size : 0

Index 3, Advertise bit 1

Route-Reflector Client

3 update-group member

RT Filter activate

Community attribute sent to this neighbor

Slow-peer detection is disabled

Slow-peer split-update-group dynamic is disabled

...

For address family: RT Filter

Session: 10.100.1.1

BGP table version 52, neighbor version 52/0

Output queue size : 0

Index 13, Advertise bit 0

Route-Reflector Client

13 update-group member

NEXT_HOP is always this router for eBGP paths

Community attribute sent to this neighbor

Default information originate, default sent

Slow-peer detection is disabled

Slow-peer split-update-group dynamic is disabled

	Sent	Rcvd
Prefix activity:	----	----
Prefixes Current:	1	2 (Consumes 160 bytes)
Prefixes Total:	1	2
Implicit Withdraw:	0	0
Explicit Withdraw:	0	0
Used as bestpath:	n/a	2
Used as multipath:	n/a	0

	Outbound	Inbound
Local Policy Denied Prefixes:	-----	-----
Bestpath from iBGP peer:	2	n/a
Total:	2	0

Number of NLRI's in the update sent: max 1, min 0
Last detected as dynamic slow peer: never
Dynamic slow peer recovered: never
Refresh Epoch: 1
Last Sent Refresh Start-of-rib: never
Last Sent Refresh End-of-rib: never
Last Received Refresh Start-of-rib: never
Last Received Refresh End-of-rib: never

	Sent	Rcvd
Refresh activity:	----	----
Refresh Start-of-RIB	0	0
Refresh End-of-RIB	0	0

Address tracking is enabled, the RIB does have a route to 10.100.1.1
Connections established 16; dropped 15
Last reset 00:14:28, due to Peer closed the session of session 1
Transport(tcp) path-mtu-discovery is enabled
Graceful-Restart is disabled

PE

```
debug bgp all
```

```
BGP: 10.100.1.3 active rcvd OPEN w/ optional parameter type 2 (Capability) len 6  
BGP: 10.100.1.3 active OPEN has CAPABILITY code: 1, length 4  
BGP: 10.100.1.3 active OPEN has MP_EXT CAP for afi/safi: 1/132  
BGP: 10.100.1.3 accept RTC SAFI
```

```
PE1# show bgp rtfiler unicast rt 1:1
```

```
BGP routing table entry for 1:2:1:1, version 3  
Paths: (1 available, best #1)  
Advertised to update-groups:  
  13  
Refresh Epoch 1  
Local  
  0.0.0.0 from 0.0.0.0 (10.100.1.1)  
    Origin IGP, localpref 100, weight 32768, valid, sourced, local, best  
    RT generation: import  
    rx pathid: 0, tx pathid: 0x0
```

O filtro AF também usa grupos de atualização:

```
PE1# show bgp rtfiler unicast all update-group 13
```

```
BGP version 4 update-group 13, internal, Address Family: RT Filter  
BGP Update version : 12/0, messages 0  
Extended-community attribute sent to this neighbor  
Topology: global, highest version: 12, tail marker: 12  
Format state: Current working (OK, last not in list)  
                Refresh blocked (not in list, last not in list)  
Update messages formatted 1, replicated 1, current 0, refresh 0, limit 1000  
Number of NLRI's in the update sent: max 2, min 0  
Minimum time between advertisement runs is 0 seconds  
Has 1 member:  
  10.100.1.3
```

Verifique o RTFilter enviado pelo PE:

```

PE1# show bgp rtfiler unicast all neighbors 10.100.1.3 advertised-routes
BGP table version is 8, local router ID is 10.100.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1:2:1:1	0.0.0.0			32768	i
*> 1:2:1:2	0.0.0.0			32768	i

Total number of prefixes 2

A codificação do prefixo de associação de destino de rota é de 4 bytes para o número de sistema autônomo e de 8 bytes para o destino de rota, que é um atributo de comunidade estendida. No exemplo acima, o prefixo de filtro de rota "1:2:1:1" é decodificado da seguinte forma:

- 1 é o número do sistema autônomo
- 2 é o tipo e o subtipo do atributo de comunidade estendida (em decimal) (consulte RFC 4360)
- 1:1 é o próprio destino da rota

O RR envia o filtro padrão para o PE (RR-client). Isso porque, por projeto, o RR deseja todas as rotas VPNv4:

```

BGP(10): (base) 10.100.1.1 send UPDATE (format) 0:0:0:0, next 10.100.1.3,
metric 0, path Local

```

O PE recebe e instala um filtro rt padrão. Por exemplo, ele envia tudo para o RR:
(debug bgp rtfiler unicast updates)

```

BGP(10): 10.100.1.3 rcvd UPDATE w/ attr: nexthop 10.100.1.3, origin i,
localpref 100, metric 0, community no-export
BGP(10): 10.100.1.3 rcvd 0:0:0:0
BGP(4): Default RT filter installed for 10.100.1.3

```

O RR recebe e instala o filtro do PE1:
(debug bgp rtfiler unicast updates)

```

BGP(10): 10.100.1.1 rcvd UPDATE w/ attr: nexthop 10.100.1.1, origin i,
localpref 100, metric 0
BGP(10): 10.100.1.1 rcvd 1:2:1:1
BGP(4): 1:2:1:1 RT filter installed for 10.100.1.1
BGP: installing rt filter on 10.100.1.1
BGP: add installed RT filter 1:2:1:1 for 10.100.1.1
BGP(10): 10.100.1.1 rcvd 1:2:1:2
BGP(4): 1:2:1:2 RT filter installed for 10.100.1.1
BGP(4): 1:2:1:2 Initiating an incremental table walk for 10.100.1.1
BGP: installing rt filter on 10.100.1.1
BGP: add installed RT filter 1:2:1:2 for 10.100.1.1

```

Verifique os filtros recebidos no RR:

```

RR1# show bgp vpnv4 unicast all neighbors 10.100.1.1 received rtfilters
Address family: VPNv4 Unicast
Extended community filter has: 2 entries with default filtering disabled
Incremental refresh walk mode
Status codes: * valid, S Stale > installed
Route-Target Outbound Filter

```

```
*> Extended Community RT:1:2
*> Extended Community RT:1:1
```

O PE não instala um filtro RT com RTs específicas. O PE recebeu o filtro rt padrão do RR, de modo que o PE envia todos os prefixos de VPNv4/v6:

```
PE1# show bgp vpnv4 unicast all neighbors 10.100.1.3 received rtfilters
Address family: VPNv4 Unicast
Extended community filter has: 1 entries with default filtering enabled
Incremental refresh walk mode
```

Para criar um filtro RT padrão, configure "neighbor x.x.x.x default-originate" em AF filter.

Isso será criado automaticamente no RR para os peerings do cliente RR.

RR

```
router bgp 1

address-family rtfiler unicast
neighbor 10.100.1.1 activate
neighbor 10.100.1.1 send-community both
neighbor 10.100.1.1 route-reflector-client
neighbor 10.100.1.1 default-originate
exit-address-family
```

Tratamento de Atualização de Rotas

Quando uma nova importação de RT é configurada ou quando a importação de RT é removida, uma atualização de rota é enviada do PE para o RR para as famílias de endereços VPNv4/6.

Quando um novo VRF é configurado, o PE envia uma atualização de rota ao RR.

Em ambos os casos com o RTC ativo, o RR não envia todos os prefixos VPNv4/6 para o PE. Ele envia somente o conjunto de acordo com o filtro RT.

Informações Relacionadas

- [Suporte Técnico e Documentação - Cisco Systems](#)