

Obstáculo de la blanco de la ruta

Contenido

[Introducción](#)

[Propósito del obstáculo de la blanco de la ruta](#)

[Comportamiento sin el RTC](#)

[Configuración de RTC](#)

[Configuración de PE](#)

[Configuración RR](#)

[Comportamiento del RTC](#)

[PE](#)

[RR](#)

[La ruta restaura la dirección](#)

[Información Relacionada](#)

Introducción

Este documento describe un mecanismo por el que el intercambio del VPNv4 y de los prefijos VPNv6 hacia el Routers del borde del proveedor (PE) esté reducido al necesario mínimo.

Propósito del obstáculo de la blanco de la ruta

Con el Multiprotocol Label Switching (MPLS) VPN, el par o el reflector de ruta (RR) del Internal Border Gateway Protocol (iBGP) envía todos los prefijos VPN4 y/o VPN6 al Routers PE. El router PE cae los prefijos VPN4/6 para los cuales no hay VPN Routing and Forwarding de importación (VRF). Esto es un comportamiento donde el RR envía los prefijos VPN4/6 al router PE, que no necesita. Ésta es una pérdida de la potencia de procesamiento en el RR y el PE y de pérdida de ancho de banda.

Con el obstáculo de la blanco de la ruta (RTC), el RR envía solamente los prefijos queridos VPN4/6 al PE. "Quiso" significa que el PE tiene VRF que importa los prefijos específicos.

El RFC 4684 especifica el RTC. El soporte está a través de un nuevo rfilter de la familia del direccionamiento para el VPNv4 y VPNv6.

El filtrado de información de la blanco de la ruta (RT) se obtiene de la lista de importación VPN RT de todos los VRF en el router PE. El router PE envía este filtrado de información como actualización de BGP en el rfilter de la familia del direccionamiento al RR. Este filtrado de información o la calidad de miembro RT se codifica en la Información de alcance de la capa de red (NLRI) de los atributos MP_REACH_NLRI y MP_UNREACH_NLRI.

El peer BGP de recepción traduce este NLRI a un filtro y instala este filtro saliente al par de envío. El peer BGP de recepción utiliza este filtro para decidir qué prefijos VPNv4/6 a enviar o a no enviar, dependiente sobre la presencia de RT asociados.

Para que el RTC trabaje, ambos peeres BGP necesitan soportar el RTC. Es decir, el RR y el PE necesitan soportarlo. Sin embargo, el despliegue puede ser ampliado, que significa que va no todo el Routers RR y PE necesita soportarlo en uno. El RTC puede trabajar en la red, con un poco de Routers PE soportando la y otras no. En el Routers que lo soporta, el RTC será ya activo. En el Routers que todavía no lo soporta los anuncios trabajarán como antes, que está sin el RTC (tan sin cualquier filtrado de salida).

Esta figura muestra el principio de RTC:

Comportamiento sin el RTC

El RR envía todos los prefijos VPN4/6 al PE. El PE cae los para las cuales no hay importación del RT. Las actualizaciones de BGP del debug muestran los prefijos caídos. NEGADA del mensaje “debido a: dan la comunidad ampliada no soportada”.

Un ejemplo para el unicast del VPNv4 es como sigue:

```
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;
```

Un ejemplo para el unicast VPNv6 es como sigue:

```
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;
```

Configuración de RTC

Configuración de 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

```

Configuración 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
```

Comportamiento del RTC

Cuando el peering BGP establece, los pares intercambian la capacidad para el rtfiler, que es 1/132 (para el VPNv4 y 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 NLRIs 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

```

El rfilter AF también utiliza los grupos de la actualización:

```

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 NLRIs in the update sent: max 2, min 0
Minimum time between advertisement runs is 0 seconds
Has 1 member:
  10.100.1.3

```

Verifique el RTFilter enviado por el 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

La codificación del prefijo de la calidad de miembro de la blanco de la ruta es 4 bytes para el número del sistema autónomo y 8 bytes para la blanco de la ruta, que es un atributo de la comunidad ampliada. En el ejemplo anterior, el prefijo el "1:2:1:1" del rtfiler está decodificado como sigue:

- 1 es el número del sistema autónomo
- 2 es el tipo y el subtipo del atributo de la comunidad ampliada (en el decimal) (refiera al RFC 4360)
- 1:1 es la blanco sí mismo de la ruta

El RR envía el filtro predeterminado a PE (RR-cliente). Esto es porque por el diseño, el RR quiere todas las rutas del VPNv4:

```

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

```

El PE recibe y instala un filtro rt del valor por defecto. Por ejemplo, envía todo al RR: (actualizaciones del unicast del rtfiler BGP del debug)

```

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

**El RR recibe y instala el rfilter del PE1:
(actualizaciones del unicast del rfilter BGP del debug)**

```
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
```

Marque los filtros recibidos en el 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
```

El PE no instala un filtro RT con los RT específicos. El PE recibió el filtro rt del valor por defecto del RR, así que el PE envía todos los prefijos 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 crear un filtro del valor por defecto RT, la configuración “vecino x.x.x.x valor por defecto-origina” bajo rfilter AF.

Esto será creada automáticamente en el RR para los peerings del cliente RR.

RR

```
router bgp 1

address-family rfilter 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
```

La ruta restaura la dirección

Cuando se configura una nueva importación RT o cuando se quita la importación RT, una ruta restaura se envía del PE al RR para las familias VPNv4/6 del direccionamiento.

Cuando se configura un nuevo VRF, el PE envía una ruta-restauración al RR.

En ambos casos con el active RTC, el RR no envía todos los prefijos VPNv4/6 al PE. Envía solamente el conjunto según el filtro RT.

Información Relacionada

- [Soporte Técnico y Documentación - Cisco Systems](#)