

Vincolo destinazione route

Sommario

[Introduzione](#)

[Scopo del vincolo di destinazione del ciclo di lavorazione](#)

[Comportamento senza RTC](#)

[Configurazione RTC](#)

[Configurazione PE](#)

[Configurazione RR](#)

[Comportamento di RTC](#)

[PE](#)

[RR](#)

[Gestione aggiornamento cicli di lavorazione](#)

[Informazioni correlate](#)

Introduzione

Questo documento descrive un meccanismo grazie al quale lo scambio di prefissi VPNv4 e VPNv6 verso router Provider Edge (PE) viene ridotto al minimo necessario.

Scopo del vincolo di destinazione del ciclo di lavorazione

Con la VPN MPLS (Multiprotocol Label Switching), il peer Border Gateway Protocol (iBGP) interno o il router Route Reflector (RR) invia tutti i prefissi VPN4 e/o VPN6 ai router PE. Il router PE scarta i prefissi VPN4/6 per i quali non è possibile importare il routing e l'inoltro VPN (VRF). Si tratta di un comportamento in cui RR invia prefissi VPN4/6 al router PE, che non è necessario. Si tratta di uno spreco di potenza di elaborazione su RR e PE e di uno spreco di larghezza di banda.

Con il vincolo di destinazione della route (RTC), il record di risorse invia solo i prefissi VPN4/6 desiderati al file PE. 'Wanted' indica che il file PE dispone di VRF che importa i prefissi specifici.

RFC 4684 specifica RTC. Il supporto avviene tramite un nuovo filtro rtfamiglia di indirizzi sia per VPNv4 che per VPNv6.

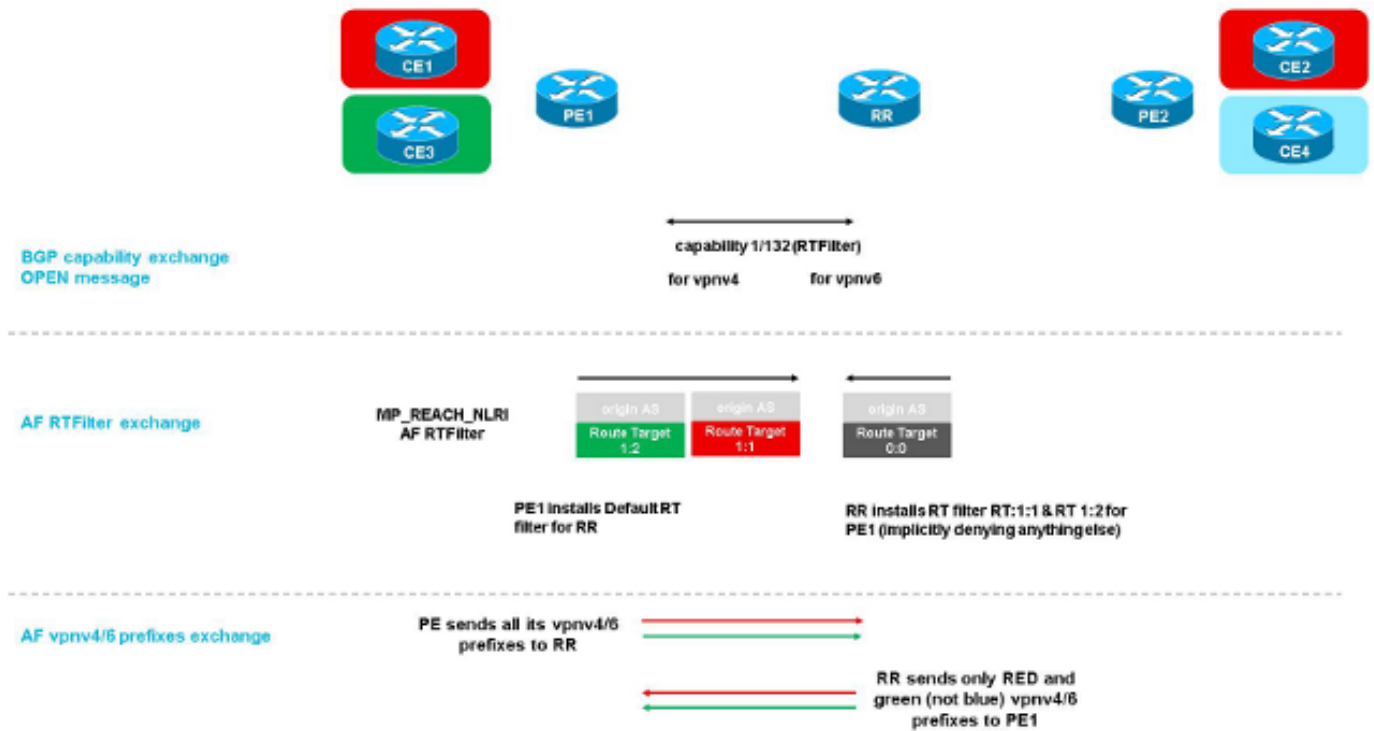
Le informazioni sul filtro Route Target (RT) vengono ottenute dall'elenco di importazione VPN RT da tutti i VRF sul router PE. Il router PE invia queste informazioni di filtro come aggiornamento BGP nella famiglia di indirizzi rfilter all'RR. Queste informazioni di filtraggio o l'appartenenza RT sono codificate nelle informazioni NLRI (Network Layer Reachability Information) degli attributi MP_REACH_NLRI e MP_UNREACH_NLRI.

Il peer BGP ricevente converte l'NLRI in un filtro e installa il filtro in uscita nel peer di invio. Il peer

BGP ricevente utilizza questo filtro per decidere quali prefissi VPNv4/6 inviare o meno, a seconda della presenza di RT collegati.

Affinché RTC funzioni, entrambi i peer BGP devono supportare RTC. In altre parole, l'RR e il PE devono sostenerlo. Tuttavia, l'installazione può essere incrementale, il che significa che non tutti i router RR e PE devono supportarla contemporaneamente. RTC può funzionare in rete, con alcuni router PE che la supportano e altri no. Sui router che lo supportano, RTC sarà già attivo. Sui router che non lo supportano ancora, gli annunci funzioneranno come prima, ossia senza RTC (quindi senza alcun filtro in uscita).

La figura mostra il principio di RTC:



Comportamento senza RTC

RR invia tutti i prefissi VPN4/6 al server PE. Il sistema PE scarta quelli per i quali non è disponibile l'importazione dell'RT. Gli aggiornamenti BGP di debug mostrano i prefissi eliminati. Il messaggio 'NEGATO a causa di: community estesa non supportata.

Di seguito è riportato un esempio di unicast VPNv4:

```
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;
```

Di seguito è riportato un esempio di unicast VPNv6:

```
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;
```

Configurazione RTC

Configurazione 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
```

Configurazione 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 di RTC

Quando il peering BGP viene stabilito, i peer si scambiano la funzionalità per rtfiler, ovvero 1/132 (per 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 multiseession 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
    Multiseession 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 rtfilter 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
```

AF rtfilter usa anche gruppi di aggiornamento:

```
PE1# show bgp rtfilter 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
```

Verificare il filtro RTF inviato dal PE:

```
PE1# show bgp rtfilter 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 codifica del prefisso di appartenenza dell'oggetto route è di 4 byte per il numero di sistema autonomo e di 8 byte per l'oggetto route, che è un attributo della community estesa. Nell'esempio precedente, il prefisso "1:2:1:1" di rfilter è decodificato come segue:

- 1 è il numero di sistema autonomo
- 2 è il tipo e il sottotipo dell'attributo della community estesa (in decimale) (fare riferimento a RFC 4360)
- 1:1 è la destinazione della route stessa

Il record di risorse invia il filtro predefinito a PE (client di risorse). Infatti, per sua progettazione, il registro di routing vuole tutte le route VPNv4:

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

Il sistema PE riceve e installa un filtro rt predefinito. Ad esempio, invia tutto all'RR:
(debug bgp rfilter 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
```

RR riceve e installa rfilter da PE1:
(debug bgp rfilter 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
```

Controllare i filtri ricevuti su RR:

```
RR1# show bgp vpnv4 unicast all neighbors 10.100.1.1 received rfilters  
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

Il PE non installa un filtro RT con RT specifici. Il sistema PE ha ricevuto il filtro rt predefinito dal server RR, quindi il sistema PE invia tutti i prefissi 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
```

Per creare un filtro RT predefinito, configurare "neighbor x.x.x.x default-originate" in AF rfilter.

Verrà creata automaticamente nel record di risorse per i peering del client di risorse.

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
```

Gestione aggiornamento cicli di lavorazione

Quando viene configurata una nuova importazione RT o quando l'importazione RT viene rimossa, viene inviato un aggiornamento di route dal PE al record di risorse per le famiglie di indirizzi VPNv4/6.

Quando viene configurato un nuovo VRF, il sistema PE invia un aggiornamento della route al record di risorse.

In entrambi i casi con RTC attivo, RR non invia tutti i prefissi VPNv4/6 al server PE. Invia il set solo in base al filtro RT.

Informazioni correlate

- [Documentazione e supporto tecnico – Cisco Systems](#)