

# Route Target-Einschränkung

## Inhalt

[Einführung](#)

[Zweck der Route Target-Einschränkung](#)

[Verhalten ohne RTC](#)

[RTC-Konfiguration](#)

[PE-Konfiguration](#)

[RR-Konfiguration](#)

[Verhalten von RTC](#)

[PE](#)

[RR](#)

[Behandlung von Routenaktualisierungen](#)

[Zugehörige Informationen](#)

## Einführung

Dieses Dokument beschreibt einen Mechanismus, bei dem der Austausch von VPNv4- und VPNv6-Präfixen zu Provider Edge (PE)-Routern auf das erforderliche Minimum reduziert wird.

## Zweck der Route Target-Einschränkung

Mit dem MPLS-VPN (Multiprotocol Label Switching) sendet der interne iBGP-Peer oder der RR (Route Reflector) alle VPN4- und/oder VPN6-Präfixe an die PE-Router. Der PE-Router verwirft die VPN4/6-Präfixe, für die kein VPN-Routing und -Forwarding (VRF) importiert wird. Dies ist ein Verhalten, bei dem der RR VPN4/6-Präfixe an den PE-Router sendet, die er nicht benötigt. Dies ist eine Verschwendung von Verarbeitungsleistung für RR und PE und eine Verschwendung von Bandbreite.

Mit Route Target Constraint (RTC) sendet der RR nur gewünschte VPN4/6-Präfixe an den PE. 'Gesucht' bedeutet, dass der PE über eine VRF-Instanz verfügt, die die spezifischen Präfixe importiert.

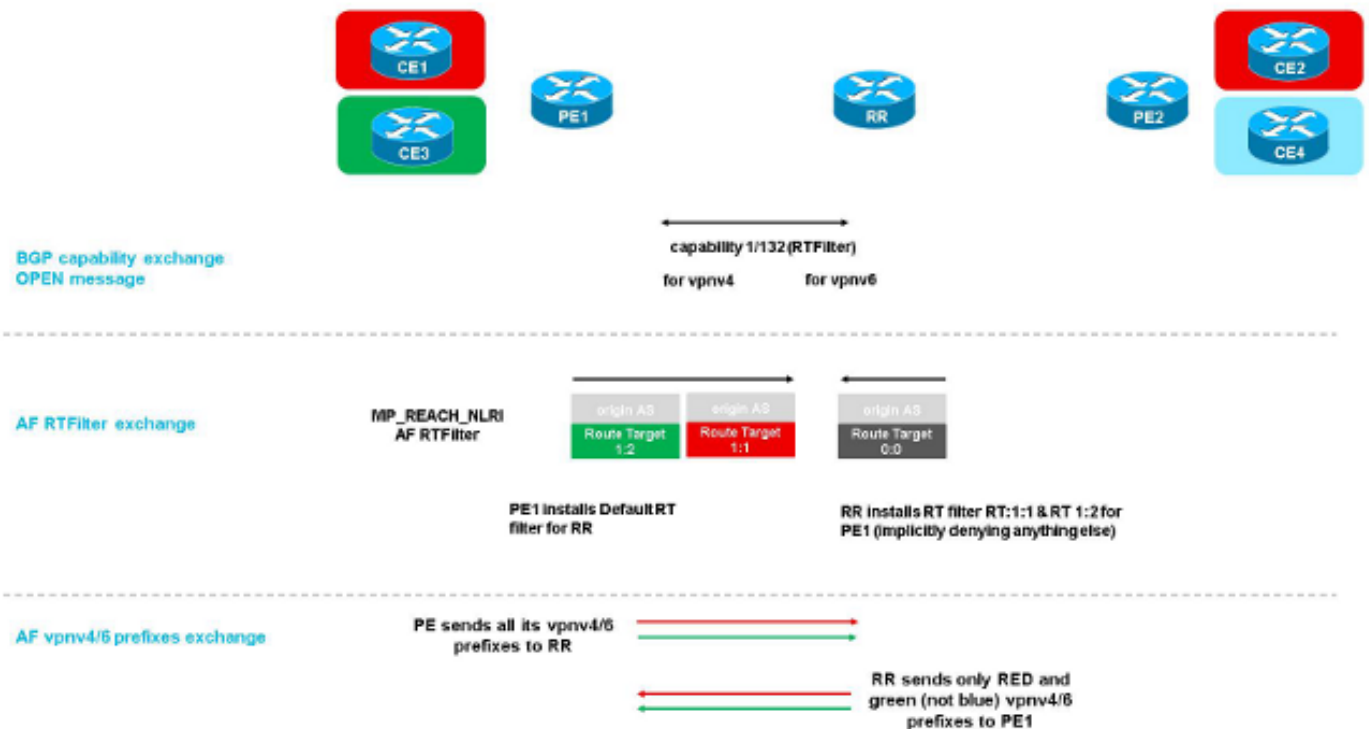
RFC 4684 gibt RTC an. Die Unterstützung erfolgt über einen neuen Adressfamilienfilter für VPNv4 und VPNv6.

Die Informationen zur Route Target (RT)-Filterung werden aus der VPN RT-Importliste aller VRFs auf dem PE-Router abgerufen. Der PE-Router sendet diese Filterinformationen als BGP-Update im Adressfamilienfilter an den RR. Diese Filterinformationen oder die RT-Mitgliedschaft werden in den Network Layer Reachability Information (NLRI) der Attribute MP\_REACH\_NLRI und MP\_UNREACH\_NLRI kodiert.

Der empfangende BGP-Peer übersetzt diesen NLRI in einen Filter und installiert diesen Filter ausgehend an den sendenden Peer. Der empfangende BGP-Peer verwendet diesen Filter, um zu entscheiden, welche VPNv4/6-Präfixe gesendet oder nicht gesendet werden sollen, abhängig von den vorhandenen angeschlossenen RTs.

Damit RTC funktioniert, müssen beide BGP-Peers RTC unterstützen. Das heißt, RR und PE müssen diese unterstützen. Die Bereitstellung kann inkrementell erfolgen, d. h., nicht alle RR- und PE-Router müssen sie in einem Schritt unterstützen. RTC kann im Netzwerk verwendet werden, wobei einige PE-Router diese unterstützen, andere jedoch nicht. Auf den Routern, die diese unterstützen, ist RTC bereits aktiv. Auf den Routern, die diese noch nicht unterstützen, funktionieren die Anzeigen wie bisher, d. h. ohne RTC (also ohne ausgehende Filterung).

Diese Abbildung zeigt das Prinzip von RTC:



## Verhalten ohne RTC

Der RR sendet alle VPNv4/6-Präfixe an den PE. Der PE verwirft diejenigen, für die kein RT importiert wird. Debug-BGP-Updates zeigen die verworfenen Präfixe an. Die Meldung 'ABGELEHNT wegen: erweiterte Community nicht unterstützt' wird gegeben.

Ein Beispiel für VPNv4-Unicast ist:

```
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;
```

Ein Beispiel für VPNv6-Unicast ist:

```
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;
```

## RTC-Konfiguration

### PE-Konfiguration

```
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
```

## RR-Konfiguration

```
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
```

## Verhalten von RTC

Wenn das BGP-Peering eingerichtet ist, tauschen die Peers die Funktion für rtfiler aus (1/132 für VPNV4 und 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 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-Filter verwendet auch Aktualisierungsgruppen:

```
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 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
```

Überprüfen Sie den vom PE gesendeten RTFilter:

```

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

Die Codierung des Präfixes für die Route Target-Mitgliedschaft beträgt 4 Byte für die autonome Systemnummer und 8 Byte für das Route Target, das ein erweitertes Community-Attribut ist. Im obigen Beispiel wird das Filterpräfix "1:2:1:1" wie folgt dekodiert:

- 1 ist die autonome Systemnummer.
- 2 ist der Typ und der Subtyp des erweiterten Community-Attributs (im Dezimalformat) (siehe RFC 4360).
- 1:1 ist das Routenziel selbst.

Der RR sendet den Standardfilter an den PE (RR-Client). Das liegt daran, dass der RR standardmäßig alle VPNv4-Routen benötigt:

```

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

```

Der PE empfängt und installiert einen Standard-rt-Filter. Beispielsweise sendet er alles an den RR:

```
(debug bgp rtfilter 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

```

Der RR empfängt und installiert den Filter von PE1:

```
(debug bgp rtfilter 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

```

Prüfen Sie die empfangenen Filter auf 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

Der PE installiert keinen RT-Filter mit bestimmten RTs. Der PE empfängt den Standard-rt-Filter des RR, sodass der PE alle VPNv4/v6-Präfixe sendet:

```
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
```

Um einen Standard-RT-Filter zu erstellen, konfigurieren Sie "neighbor x.x.x.x default-originate" unter AF rtfiler.

Diese wird automatisch auf dem RR für die RR-Client-Peerings erstellt.

## 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
```

## Behandlung von Routenaktualisierungen

Wenn ein neuer RT-Import konfiguriert oder der RT-Import entfernt wird, wird eine Routen-Aktualisierung vom PE an den RR für die Adressfamilien VPNv4/6 gesendet.

Wenn eine neue VRF-Instanz konfiguriert wird, sendet der PE eine Routen-Aktualisierung an den RR.

In beiden Fällen, in denen RTC aktiv ist, sendet der RR nicht alle VPNv4/6-Präfixe an den PE. Es sendet nur das Set gemäß RT-Filter.

## Zugehörige Informationen

- [Technischer Support und Dokumentation - Cisco Systems](#)