

# Route Target Constraint



Document ID: 116062

Contributed by Luc De Ghein, Cisco TAC Engineer.  
Apr 30, 2013

## Contents

### Introduction

### Purpose of Route Target Constraint

### Behavior Without RTC

### RTC Configuration

- PE Configuration

- RR Configuration

### Behavior of RTC

- PE

- RR

- Route Refresh Handling

### Related Information

## Introduction

This document describes a mechanism whereby the exchange of VPNv4 and VPNv6 prefixes towards Provider Edge (PE) routers is reduced to the minimum necessary.

## Purpose of Route Target Constraint

With Multiprotocol Label Switching (MPLS) VPN, the internal Border Gateway Protocol (iBGP) peer or Route Reflector (RR) sends all VPN4 and/or VPN6 prefixes to the PE routers. The PE router drops the VPN4/6 prefixes for which there is no importing VPN routing and forwarding (VRF). This is a behavior where the RR sends VPN4/6 prefixes to the PE router, which it does not need. This is a waste of processing power on the RR and the PE and a waste of bandwidth.

With Route Target Constraint (RTC), the RR sends only wanted VPN4/6 prefixes to the PE. 'Wanted' means that the PE has VRF importing the specific prefixes.

RFC 4684 specifies RTC. The support is through a new address family rtfiler for both VPNv4 and VPNv6.

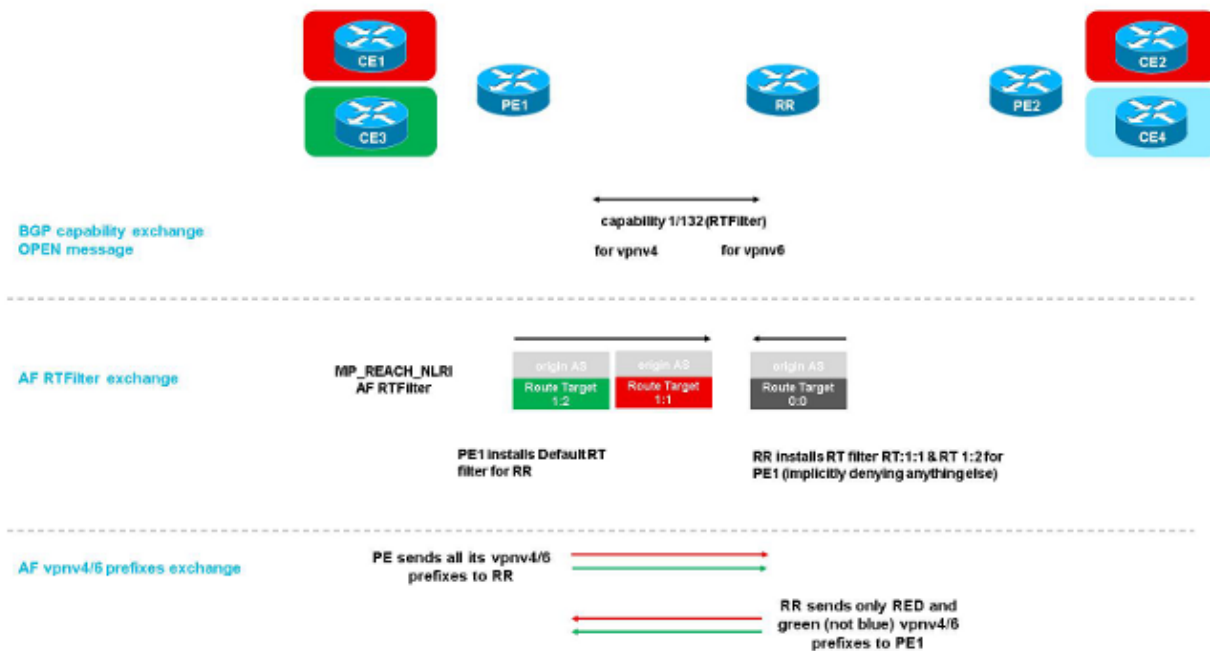
The Route Target (RT) filtering information is obtained from the VPN RT import list from all the VRFs on the PE router. The PE router sends this filtering information as a BGP update in the address family rtfiler to the RR. This filtering information or RT membership is encoded in the Network Layer Reachability Information (NLRI) of the MP\_REACH\_NLRI and MP\_UNREACH\_NLRI attributes.

The receiving BGP peer translates this NLRI into a filter and installs this filter outbound to the sending peer. The receiving BGP peer uses this filter to decide which VPNv4/6 prefixes to send or not send, dependent upon the presence of attached RTs.

For RTC to work, both BGP peers need to support RTC. That is, the RR and the PE need to support it. However, the deployment can be incremental, which means not all RR and PE routers need to support it in one go. RTC can work in the network, with some PE routers supporting it and others not. On the routers that support it, RTC will already be active. On the routers that do not yet support it the advertisements will work

as before, which is without RTC (so without any outbound filtering).

This figure shows the principle of RTC:



## Behavior Without RTC

The RR sends all VPN4/6 prefixes to the PE. The PE drops the ones for which there is no import of the RT. Debug BGP updates show the dropped prefixes. The message 'DENIED due to: extended community not supported' is given.

An example for VPNv4 unicast is as follows:

```
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;
```

An example for VPNv6 unicast is as follows:

```
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 Configuration

## PE Configuration

```
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 Configuration

```
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
```

# Route Target Constraint

## Behavior of RTC

When the BGP peering establishes, the peers exchange the capability for rtfiler, which is 1/132 (for VPNv4 and 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 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
```

The AF rfilter also uses update groups:

```
PE1# show bgp rfilter 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
```

Verify the RTFilter sent by the PE:

```
PE1# show bgp rfilter 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
```

The encoding of the Route Target Membership Prefix is 4 bytes for the Autonomous System Number and 8 bytes for the Route Target, which is an extended community attribute. In the example above, the rfilter prefix "1:2:1:1" is decoded as follows:

- 1 is the Autonomous System Number
- 2 is the type and subtype of the extended community attribute (in decimal) (refer to RFC 4360)
- 1:1 is the Route Target itself

The RR sends the default filter to PE (RR-client). This is because by design, the RR wants all the VPNv4 routes:

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

The PE receives and installs a default rt filter. For example, it sends everything to the 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
```

The RR receives and installs the rfilter from 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
```

Check the received filters on 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
```

The PE does not install an RT filter with specific RTs. The PE received the default rt filter from the RR, so the PE sends all VPNv4/v6 prefixes:

```
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
```

In order to create a default RT filter, configure "neighbor x.x.x.x default-originate" under AF rtfiler.

This will be created automatically on the RR for the RR client peerings.

## 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
```

## Route Refresh Handling

When a new RT import is configured or when the RT import is removed, a route refresh is sent from the PE to the RR for the address families VPNv4/6.

When a new VRF is configured, the PE sends a route-refresh to the RR.

In both cases with RTC active, the RR does not send all VPNv4/6 prefixes to the PE. It only sends the set according to the RT filter.



# Related Information

- *Technical Support & Documentation – Cisco Systems*

---

Updated: Apr 30, 2013

Document ID: 116062

---