BGP Event-Based VPN Import

The BGP Event-Based VPN Import feature introduces a modification to the existing Border Gateway Protocol (BGP) path import process. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available. Convergence times are significantly reduced because there is no longer any delay in the propagation of routes due to the software waiting for a periodic scanner time interval before processing the updates. To implement the new processing, new command-line interface (CLI) commands are introduced.

- Finding Feature Information, page 1
- Prerequisites for BGP Event-Based VPN Import, page 1
- Information About BGP Event-Based VPN Import, page 2
- How to Configure BGP Event-Based VPN Import, page 3
- Configuration Examples for BGP Event-Based VPN Import, page 9
- Additional References, page 10
- Feature Information for BGP Event-Based VPN Import, page 10

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for BGP Event-Based VPN Import

Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled on all participating routers.
Information About BGP Event-Based VPN Import

BGP Event-BasedVPN Import

The BGP Event-Based VPN Import feature introduces a modification to the existing BGP path import process. BGP Virtual Private Network (VPN) import provides importing functionality for BGP paths where BGP paths are imported from the BGP VPN table into a BGP virtual routing and forwarding (VRF) topology. In the existing path import process, when path updates occur, the import updates are processed during the next scan time which is a configurable interval of 5 to 15 seconds. The scan time adds a delay in the propagation of routes. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available.

Using the BGP Event-Based VPN Import feature, convergence times are significantly reduced because provider edge (PE) routers can propagate VPN paths to customer edge (CE) routers without the scan time delay. Configuration changes such as adding imported route-targets to a VRF are not processed immediately, and are still handled during the 60-second periodic scanner pass.

Import Path Selection Policy

The BGP Event-Based VPN Import feature introduces three path selection policies:

- **All**—Import all available paths from the exporting net that match any Route Target (RT) associated with the importing VRF instance.
- **Bestpath**—Import the best available path that matches the RT of the VRF instance. If the bestpath in the exporting net does not match the RT of the VRF instance, a best available path that matches the RT of the VRF instance is imported.
- **Multipath**—Import the bestpath and all paths marked as multipaths that match the RT of the VRF instance. If there are no bestpath or multipath matches, then the best available path is selected.

Multipath and bestpath options can be restricted using an optional keyword to ensure that the selection is made only on the configured option. If the **strict** keyword is configured, the software disables the fall back safety option of choosing the best available path. If there are no paths appropriate to the configured option (bestpath or multipath) in the exporting net that match the RT of the VRF instance, then no paths are imported. This behavior matches the behavior of the software before the BGP Event-Based VPN Import feature was introduced.

When the restriction is not set, paths that are imported as the best available path are tagged. In `show` command output these paths are identified with the wording, “imported safety path.”

The paths existing in an exporting net that are considered for import into a VRF instance may have been received from another peer router and were not subject to the VPN importing rules. These paths may contain the same route-distinguisher (RD) information because the RD information is local to a router, but some of these paths do not match the RT of the importing VRF instance and are marked as “not-in-vrf” in the `show` command output. Any path that is marked as “not-in-vrf” is not considered as a bestpath because paths not in the VRF appear less attractive than paths in the VRF.
Import Path Limit

To control the memory utilization, a maximum limit of the number of paths imported from an exporting net can be specified per importing net. When a selection is made of paths to be imported from one or more exporting net, the first selection priority is a best path, the next selection priority is for multipaths, and the lowest selection priority is for nonmultipaths.

How to Configure BGP Event-Based VPN Import

Configuring a Multiprotocol VRF

Perform this task to configure a multiprotocol VRF that allows you to share route-target policies (import and export) between IPv4 and IPv6 or to configure separate route-target policies for IPv4 and IPv6 VPNs. In this task, only the IPv4 address family is configured, but we recommend using the multiprotocol VRF configuration for all new VRF configurations.

Note

This task is not specific to the BGP Event-Based VPN Import feature.

SUMMARY STEPS

1. enable
2. configure terminal
3. vrf definition vrf-name
4. rd route-distinguisher
5. route-target {import | export | both} route-target-ext-community
6. address-family ipv4 [unicast]
7. exit-address-family
8. exit
9. interface type number
10. vrf forwarding vrf-name
11. ip address ip-address mask
12. no shutdown
13. exit
14. Repeat Step 3 through Step 13 to bind other VRF instances with an interface.
15. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
</tr>
<tr>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

Router# configure terminal

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong> vrf definition vrf-name</td>
</tr>
<tr>
<td>Configures a VRF routing table and enters VRF configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config)# vrf definition vrf-A

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong> rd route-distinguisher</td>
</tr>
<tr>
<td>Creates routing and forwarding tables and specifies the default route distinguisher for a VPN.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-vrf)# rd 45000:1

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> route-target {import</td>
</tr>
<tr>
<td>Creates a route target extended community for a VRF.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-vrf)# route-target both 45000:100

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong> address-family ipv4 [unicast]</td>
</tr>
<tr>
<td>Specifies the IPv4 address family and enters VRF address family configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-vrf)# address-family ipv4 unicast

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong> exit-address-family</td>
</tr>
<tr>
<td>Exits VRF address family configuration mode and returns to VRF configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-vrf-af)# exit-address-family
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 8</strong> exit</td>
<td>Exits VRF configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config-vrf)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> interface  <em>type number</em></td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config)# interface FastEthernet 1/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> vrf forwarding  <em>vrf-name</em></td>
<td>Associates a VRF instance with the interface configured in Step 9.</td>
</tr>
<tr>
<td>Example: Router(config-if)# vrf forwarding vrf-A</td>
<td>- When the interface is bound to a VRF, previously configured IP addresses are removed, and the interface is disabled.</td>
</tr>
<tr>
<td><strong>Step 11</strong> ip address  <em>ip-address mask</em></td>
<td>Configures an IP address for the interface.</td>
</tr>
<tr>
<td>Example: Router(config-if)# ip address 10.4.8.149 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> no shutdown</td>
<td>Restarts a disabled interface.</td>
</tr>
<tr>
<td>Example: Router(config-if)# no shutdown</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> exit</td>
<td>Exits interface configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td>Repeat Step 3 through Step 13 to bind other VRF instances with an interface.</td>
</tr>
<tr>
<td><strong>Step 15</strong> end</td>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Router(config)# end</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Event-Based VPN Import Processing for BGP Paths

Perform this task to reduce convergence times when BGP paths change by configuring event-based processing for importing BGP paths into a VRF table. Two new CLI commands allow the configuration of a maximum number of import paths per importing net and the configuration of a path selection policy.

Before You Begin

This task assumes that you have previously configured the VRF to be used with the VRF address family syntax. To configure a VRF, see the "Configuring a Multiprotocol VRF" section earlier in this module.

Complete BGP neighbor configuration is also assumed. For an example configuration, see the "Example: Configuring Event-Based VPN Import Processing for BGP Paths" section in this module.

SUMMARY STEPS

1. enable
2. configure terminal
3. router bgp autonomous-system-number
4. address-family ipv4 vrf vrf-name
5. import path selection {all | bestpath [strict] | multipath [strict]}
6. import path limit number-of-import-paths
7. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>router bgp autonomous-system-number</td>
<td>Enters router configuration mode for the specified routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# router bgp 45000</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

<table>
<thead>
<tr>
<th>Step 4</th>
<th>address-family ipv4 vrf vrf-name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-router)# address-family ipv4 vrf vrf-A</td>
</tr>
</tbody>
</table>

**Purpose**

Specifies the IPv4 address family and enters address family configuration mode.

- Use the `vrf` keyword and `vrf-name` argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.

| Step 5 | import path selection {all | bestpath [strict] | multipath [strict]} |
|--------|------------------------|
| Example: | Router(config-router-af)# import path selection all |

**Purpose**

Specifies the BGP path selection policy for importing routes into a VRF table.

- In this example, all paths that match any RT of the VRF instance are imported.

<table>
<thead>
<tr>
<th>Step 6</th>
<th>import path limit number-of-import-paths</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-router-af)# import path limit 3</td>
</tr>
</tbody>
</table>

**Purpose**

Specifies, per importing net, a maximum number of BGP paths that can be imported from an exporting net.

<table>
<thead>
<tr>
<th>Step 7</th>
<th>end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-router-af)# end</td>
</tr>
</tbody>
</table>

**Purpose**

Exits address family configuration mode and returns to privileged EXEC mode.

---

### Monitoring and Troubleshooting BGP Event-Based VPN Import Processing

Perform the steps in this task as required to monitor and troubleshoot the BGP event-based VPN import processing.

Only partial command syntax for the **show** commands used in this task is displayed. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

**SUMMARY STEPS**

1. enable
2. show ip bgp vpnv4 {all | rd route-distinguisher | vrf vrf-name} [network-address [mask]]
3. show ip route [vrf vrf-name] [ip-address [mask]]
4. debug ip bgp vpnv4 unicast import {events | updates [access-list]}
DETAILED STEPS

**Step 1** `enable`
Enables privileged EXEC mode. Enter your password if prompted.

**Example:**
```
Router> enable
```

**Step 2** `show ip bgp vpnv4 {all | rd route-distinguisher | vrf vrf-name} [network-address [mask]]`
In this example output, a safe import path selection policy is in effect because the `strict` keyword is not configured using the `import path selection` command. When a path is imported as the best available path (when the bestpath or multipaths are not eligible for import), the path is marked with "imported safety path," as shown in the output.

**Example:**
```
Router# show ip bgp vpnv4 all 172.17.0.0
```

The paths existing in an exporting net that are considered for import into a VRF instance may have been received from another peer router and were not subject to the VPN importing rules. These paths may contain the same route-distinguisher (RD) information because the RD information is local to a router, but some of these paths do not match the RT of the importing VRF instance and are marked as "not-in-vrf" in the `show` command output.

In the following example output, a path was received from another peer router and was not subject to the VPN importing rules. This path, `10.0.101.2`, was added to the VPNv4 table and associated with the vrf-A net because it contains a match of the RD information although the RD information was from the original router. This path is not, however, an RT match for vrf-A and is marked as "not-in-vrf." Note that on the net for vrf-A, this path is not the bestpath because any paths that are not in the VRF appear less attractive than paths in the VRF.

**Example:**
```
Router# show ip bgp vpnv4 all 172.17.0.0
```

**Step 3** `show ip route [vrf vrf-name] [ip-address [mask]]`
In this example output, information about the routing table for VRF vrf-A is displayed:

Example:

Router# show ip route vrf vrf-A 172.17.0.0

Routing Table: vrf-A
Routing entry for 172.17.0.0/16
Known via "bgp 1", distance 200, metric 50
Tag 2, type internal
Last update from 10.0.101.33 00:00:32 ago
Routing Descriptor Blocks:
  * 10.0.101.33 (default), from 10.0.101.33, 00:00:32 ago
    Route metric is 50, traffic share count is 1
    AS Hops 1
    Route tag 2
    MPLS label: 16
    MPLS Flags: MPLS Required

Step 4  debug ip bgp vpnv4 unicast import {events | updates [access-list]}
Use this command to display debugging information related to the importing of BGP paths into a VRF instance table. The actual output depends on the commands that are subsequently entered.

Note  If no access list to filter prefixes is specified when using the updates keyword, all updates for all prefixes are displayed and this may slow down your network.

Example:

Router# debug ip bgp vpnv4 unicast import events

BGP import events debugging is on

Configuration Examples for BGP Event-Based VPN Import

Example: Configuring Event-Based VPN Import Processing for BGP Paths

In this example, a VRF (vrf-A) is configured and VRF forwarding is applied to Fast Ethernet interface 1/1. In address family mode, the import path selection is set to all and the number of import paths is set to 3. Two BGP neighbors are configured under the IPv4 address family and activated under the VPNv4 address family.

vrf definition vrf-A
rd 45000:1
route-target import 45000:100
address-family ipv4
  exit-address-family
!
interface FastEthernet1/1
no ip address
vrf forwarding vrf-A
ip address 10.4.8.149 255.255.255.0
no shut
exit
!
routing bgp 45000
  network 172.17.1.0 mask 255.255.255.0

Step 4  debug ip bgp vpnv4 unicast import {events | updates [access-list]}
Use this command to display debugging information related to the importing of BGP paths into a VRF instance table. The actual output depends on the commands that are subsequently entered.

Note  If no access list to filter prefixes is specified when using the updates keyword, all updates for all prefixes are displayed and this may slow down your network.

Example:

Router# debug ip bgp vpnv4 unicast import events

BGP import events debugging is on
neighbor 192.168.1.2 remote-as 40000
neighbor 192.168.3.2 remote-as 50000
address-family ipv4 vrf vrf-A
import path selection all
import path limit 3
exit-address-family
address-family vpnv4
neighbor 192.168.1.2 activate
neighbor 192.168.3.2 activate
end

**Additional References**

**Related Documents**

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
<tr>
<td>BGP commands</td>
<td>Cisco IOS IP Routing: BGP Command Reference</td>
</tr>
</tbody>
</table>

**Standards and RFCs**

<table>
<thead>
<tr>
<th>Standard/RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 2918</td>
<td>Route Refresh Capability for BGP-4</td>
</tr>
</tbody>
</table>

**Technical Assistance**

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

**Feature Information for BGP Event-Based VPN Import**

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to . An account on Cisco.com is not required.

**Table 1: Feature Information for BGP Event-Based VPN Import**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| BGP Event-Based VPN Import    | 15.0(1)SY| The BGP Event-Based VPN Import feature introduces a modification to the existing Border Gateway Protocol (BGP) path import process. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available. Convergence times are significantly reduced because there is no longer any delay in the propagation of routes due to the software waiting for a periodic scanner time interval before processing the updates. To implement the new processing, new command-line interface (CLI) commands are introduced.

The following commands were introduced or modified: `bgp scan-time`, `import path limit`, `import path selection`, `maximum-paths eibgp`, `maximum-paths ibgp`, `show ip bgp vpnv4`. |