BGP Event-Based VPN Import

Last Updated: April 13, 2012

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

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About BGP Event-Based VPN Import

BGP Event-Based VPN 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

Event-based VPN import 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 bestpath, 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

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vrf-A`	Configures a VRF routing table and enters VRF configuration mode. • Use the *vrf-name* argument to specify a name to be assigned to the VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 45000:1`	Creates routing and forwarding tables and specifies the default route distinguisher for a VPN. • Use the *route-distinguisher* argument to add an 8-byte value to an IPv4 prefix to create a unique VPN IPv4 prefix.
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target both 45000:100`	Creates a route target extended community for a VRF. • Use the **import** keyword to import routing information from the target VPN extended community. • Use the **export** keyword to export routing information to the target VPN extended community. • Use the **both** keyword to both import routing information from, and export routing information to, the target VPN extended community. • Use the *route-target-ext-community* argument to add the route target extended community attributes to the VRF's list of import, export, or both (import and export) route target extended communities.
Step 6	**address-family ipv4** [**unicast**] **Example:** `Router(config-vrf)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters VRF address family configuration mode. • This step is required here to specify an address family for the VRF defined in the previous steps.

	Command or Action	Purpose
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 1/1`	Enters interface configuration mode.
Step 10	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding vrf-A`	Associates a VRF instance with the interface configured in Step 9. • When the interface is bound to a VRF, previously configured IP addresses are removed, and the interface is disabled.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.4.8.149 255.255.255.0`	Configures an IP address for the interface.
Step 12	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Restarts a disabled interface.
Step 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 14	Repeat Step 3 through Step 13 to bind other VRF instances with an interface.	--

	Command or Action	Purpose

Step 15	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

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.

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, page 3.

Complete BGP neighbor configuration is also assumed. For an example configuration, see the Configuring Event-Based VPN Import Processing for BGP Paths Example, page 9.

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

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv4 vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv4 vrf vrf-A`	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`
		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.

Step 6	**import path limit** *number-of-import-paths*
	Example:
	`Router(config-router-af)# import path limit 3`
		Specifies, per importing net, a maximum number of BGP paths that can be imported from an exporting net.

Step 7	**end**
	Example:
	`Router(config-router-af)# end`
		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 Referenc*e.

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

BGP routing table entry for 45000:1:172.17.0.0/16, version 10
Paths: (1 available, best #1, table vrf-A)
Flag: 0x820
   Not advertised to any peer
   2, imported safety path from 50000:2:172.17.0.0/16
     10.0.101.1 from 10.0.101.1 (10.0.101.1)
       Origin IGP, metric 200, localpref 100, valid, internal, best
       Extended Community: RT:45000:100
```

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

BBGP routing table entry for 45000:1:172.17.0.0/16, version 11
Paths: (2 available, best #2, table vrf-A)
Flag: 0x820
   Not advertised to any peer
   2
     10.0.101.2 from 10.0.101.2 (10.0.101.2)
       Origin IGP, metric 100, localpref 100, valid, internal, not-in-vrf
       Extended Community: RT:45000:200
       mpls labels in/out nolabel/16
   2
     10.0.101.1 from 10.0.101.1 (10.0.101.1)
       Origin IGP, metric 50, localpref 100, valid, internal, best
       Extended Community: RT:45000:100
       mpls labels in/out nolabel/16
```

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

Configuring Event-Based VPN Import Processing for BGP Paths Example

In this example configuration, 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
!
router bgp 45000
 network 172.17.1.0 mask 255.255.255.0
 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
```

Where to Go Next

- If you want to connect to an external service provider and use other external BGP features, see the "Connecting to a Service Provider Using External BGP" module.
- If you want to configure some internal BGP features, see the "Configuring Internal BGP Features" module.
- If you want to configure BGP neighbor session options, see the "Configuring BGP Neighbor Session Options" module.
- If you want to configure some advanced BGP features, see the "Configuring Advanced BGP Features" module.

Additional References

The following sections provide references related to the BGP Event-Based VPN Import feature.

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: BGP Command Reference*
Overview of Cisco BGP conceptual information with links to all the individual BGP modules	"Cisco BGP Overview" module of the *Cisco IOS IP Routing: BGP Configuration Guide.*
Conceptual and configuration details for basic BGP tasks.	"Configuring a Basic BGP Network" module of the *Cisco IOS IP Routing Protocols Configuration Guide*.
Command Lookup Tool	http://tools.cisco.com/Support/CLILookup
Cisco IOS Master Command List	http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for BGP Event-Based VPN Import

Feature Name	Releases	Feature Information
BGP Event-Based VPN Import	12.2(33)SRE 15.0(1)M 15.0(1)S Cisco IOS XE 3.1.0SG, 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**.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.