L2VPN Advanced VPLS

Last Updated: December 23, 2011

First Published: June 4, 2010

Last Updated: June 4, 2010

The L2VPN Advanced VPLS feature introduces the following enhancements to Virtual Private LAN Services:

- Ability to load-balance traffic across multiple core interfaces using equal cost multipaths (ECMP)
- Support for redundant provide edge switches
- Command line interface enhancements to facilitate configuration of the L2VPN Advanced VPLS feature

The L2VPN Advanced VPLS feature uses Virtual Switch System (VSS) and Flow Aware Transport (FAT) pseudowires to achieve PE redundancy and load-balancing. The following sections explain the concepts and configuration tasks for this feature.

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 for L2VPN Advanced VPLS, page 15.

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

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

Contents

Prerequisites for L2VPN Advanced VPLS

- This feature requires that you understand how VPLS works. For information about VPLS, see VPLS Overview in the Cisco 7600 Series Ethernet Services Plus (ES+) and Ethernet Services Plus T (ES+T) Line Card Configuration Guide.
- Configuring the L2VPN Advanced VPLS feature works with MPLS Traffic Engineering tunnels with explicit paths and Generic Routing Encapsulation (GRE tunnels) with static routes to the tunnel destination. For information and configuration steps for MPLS traffic engineering and GRE tunnels, see the following documents:
    - MPLS Traffic Engineering and Enhancements
    - Implementing Tunnels
- This features requires two Cisco 6500 series routers be configured as a virtual switch system.
- This features requires nonstop forwarding and stateful switchover.

Restrictions for L2VPN Advanced VPLS

- The**ping** and **traceroute** commands that support the Any Transport over MPLS Virtual Circuit Connection Verification (VCCV) feature are not supported over FAT pseudowires.
- The VPLS Autodiscovery feature is not supported with the L2VPN Advanced VPLS feature.
- In Cisco IOS Release 12.2(33)SXI4, the following types of configurations are supported:
    - MPLS core with configuration of PE routers through the **neighbor** command under transport vpls mode.
    - MPLS core with configuration of PE routers through MPLS traffic engineering tunnels using explicit paths.
    - IP core with configuration of PE routers through MPLS over GRE tunnels.

Other configuration methods, including using the **route-via** command, BGP autodiscovery, or explicit VLAN assignment to a PE egress port, are not supported.

- Load-balancing is not supported in the core routers when the core uses IP to transport packets.
- The maximum number of links per bundle is limited to eight.
- The maximum number of port channels is limited to 32.
- The maximum number of VPLS neighbors is limited to 60 minus the number of neighbors configured with the **load-balanceflow** command.
- In Cisco IOS Release 12.2(33)SXI4, the L2VPN Advanced VPLS feature is supported on the Cisco Catalyst 6500 series switches with Supervisor 720-10GE engine.
- The L2VPN Advanced VPLS feature supports the following line cards and shared port adapters (SPAs):

- ◦ 7600-SIP-400 (core facing)
- ◦ Gigabit and 10-gigabit Ethernet SPAs (2X1GE-V1, 2X1GE-V2 and 1X10GE-V2 SPA)
- ◦ Packet over Sonet (POS) SPAs (2XOC3, 4XOC3, 1XOC12 and 1XOC48)

Information About L2VPN Advanced VPLS

To configure the L2VPN Advanced VPLS feature, you should understand the following concepts:

- FAT Pseudowires and Their Role in Load-Balancing, page 3
- Virtual Switch Systems, page 3

FAT Pseudowires and Their Role in Load-Balancing

FAT pseudowires are used to load-balance traffic in the core when equal cost multipaths are used. The MPLS labels add an additional label to the stack, called the flow label, which contains the flow information of a VC. For more information about FAT pseudowires, see PWE3 Internet-Draft *Flow Aware Transport of MPLS Pseudowires* (draft-bryant-filsfils-fat-pw).

Virtual Switch Systems

Two Cisco 6500 series switches can be connected to form one logical switch. One switch is designated as the master, while the other is the slave. The two switches are connected by a virtual switch link (VSL). The two switches are used for link redundancy, load-balancing, and failover.

For more information on virtual switch systems, see Configuring VSS in the *Catalyst 6500 Release 12.2SXH and Later Software Configuration Guide*

How to Configure L2VPN Advanced VPLS

The following sections explain how to configure the L2VPN Advanced VPLS feature:

- Enabling Load-Balancing with ECMP and FAT Pseudowires, page 3
- Enabling Port-Channel Load-Balancing, page 5
- Explicitly Specifying the PE Routers As Part of Virtual Ethernet Interface Configuration, page 6
- Configuring an MPLS Traffic Engineering Tunnel, page 8
- Configuring a GRE Tunnel, page 10

Enabling Load-Balancing with ECMP and FAT Pseudowires

The following steps explain how to enable load-balancing at the provider edge (PE) routers and on the core routers.

To enable load-balancing on the edge routers, issue the **load-balanceflow** command. The load-balancing rules are configured through the **port-channelload-balance** command parameters.

To enable core load-balancing, issue the **flow-labelenable** command on both PE routers. You must issue the **load-balanceflow** command with the **flow-labelenable** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *name*
4. **encapsulation mpls**
5. **load-balance flow**
6. flow-label enable
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.
Step 3	**pseudowire-class** *name* **Example:** `Router(config)# pseudowire-class class1`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw)# encapsulation mpls`	Specifies the MPLS tunneling encapsulation type.
Step 5	**load-balance flow** **Example:** `Router(config-pw)# load-balance flow`	Enables load-balancing on ECMPs.

	Command or Action	Purpose
Step 6	flow-label enable **Example:** `Router(config-pw)# flow-label enable`	Enables the imposition and disposition of flow labels for the pseudowire.

Step 7	end
	Example:
	`Router(config-pw)# end`
		Exits pseudowire class configuration mode and enters privileged EXEC mode.

Enabling Port-Channel Load-Balancing

The following task explains how to enable port channel load-balancing, which sets the load-distribution method among the ports in the bundle. If the **port-channelload-balance** command is not configured, load-balancing occurs with default parameters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **port-channel load-balance** *method*
4. exit

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	**port-channel load-balance** *method* **Example:** `Router(config)# port-channel load-balance src-mac`	Specifies the load distribution method among the ports in a bundle.
Step 4	exit **Example:** `Router(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

Explicitly Specifying the PE Routers As Part of Virtual Ethernet Interface Configuration

There are several ways to specify the route through which traffic should pass.

- Explicitly specify the PE routers as part of the virtual Ethernet interface configuration
- Configure an MPLS Traffic Engineering tunnel
- Configure a GRE tunnel

The following task explains how to explicitly specify the PE routers as part of the virtual Ethernet interface configuration.

Note: This tasks includes steps for configuring the LAN port for Layer 2 Switching. For more information, see Configuring LAN Ports for Layer 2 Switching.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-ethernet** *num*
4. **transport vpls mesh**
5. **neighbor** *remote-router-id* [**pw-class** pw-class-name]
6. exit
7. switchport
8. switchport mode trunk
9. **switchport trunk allowed vlan** {**add** | **except** | **none** | **remove**} *vlan* [,*vlan*[,*vlan*[,...]]
10. exit

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	**interface virtual-ethernet** *num* **Example:** `Router(config)# interface virtual-ethernet 1`	Creates a virtual Ethernet interface and enters interface configuration mode.
Step 4	**transport vpls mesh** **Example:** `Router(config-if)# transport vpls mesh`	Create a full mesh of pseudowires and enters VPLS transport mode.
Step 5	**neighbor** *remote-router-id* [**pw-class** pw-class-name] **Example:** `Router(config-if-transport)# neighbor 10.19.19.19 pw-class 1`	Specifies the PE routers to be used in the pseudowire.
Step 6	exit **Example:** `Router(config-if-transport)# exit`	Exits VPLS transport configuration mode and enters interface configuration mode.
Step 7	switchport **Example:** `Router(config-if)# switchport`	Configures the port for Layer 2 switching.

	Command or Action	Purpose
Step 8	switchport mode trunk **Example:** `Router(config-if)# switchport mode trunk`	Enables permanent trunking mode and negotiates to convert the link into a trunk link.
Step 9	**switchport trunk allowed vlan** {**add** | **except** | **none** | **remove**} *vlan* [,*vlan*[,*vlan*[,...]] **Example:** `Router(config-if)# switchport trunk allowed vlan 10, 20`	Configures the list of VLANs allowed on the trunk.
Step 10	exit **Example:** `Router(config)# exit`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring an MPLS Traffic Engineering Tunnel

There are several ways to specify the route through which traffic should pass.

- Explicitly specify the PE routers as part of the virtual Ethernet interface configuration
- Configure an MPLS Traffic Engineering tunnel
- Configure a GRE tunnel

The following task explains how to configure an MPLS Traffic Engineering tunnel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip unnumbered** *type number*
5. **tunnel destination** *ip-address*
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng autoroute announce**
8. **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name***path-name*} | **identifier***path-number*} [**lockdown**]
9. **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	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel10`	Configures an interface type and enters interface configuration mode.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered loopback 0`	Assigns an IP address to the tunnel interface. - An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 10.20.1.1`	Specifies the destination for a tunnel. - The *ip-address* keyword is the IP address of the host destination expressed in dotted decimal notation.
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Configures the tunnel encapsulation mode to MPLS traffic engineering.
Step 7	**tunnel mpls traffic-eng autoroute announce** **Example:** `Router(config-if)# tunnel mpls traffic-eng autoroute announce`	Configures the IGP to use the tunnel in its enhanced SPF calculation.

	Command or Action	Purpose
Step 8	**tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name***path-name*} | **identifier***path-number*} [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 1 explicit identifier 1`	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. • A dynamic path is used if an explicit path is currently unavailable.

Step 9	**end**
	Example:
	`Router(config-if)# exit`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuring a GRE Tunnel

There are several ways to specify the route through which traffic should pass.

- Explicitly specify the PE routers as part of the virtual Ethernet interface configuration
- Configure an MPLS Traffic Engineering tunnel
- Configure a GRE tunnel

The following task explains how to configure a GRE tunnel. For more information on GRE tunnels, see Implementing Tunnels.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **tunnel mode** {**greip**|**gremultipoint**}
5. **mpls ip**
6. **tunnel source** {*ip-address* | *interface-typeinterface-number*}
7. **tunnel destination** {*hostname* | *ip-address*}
8. **end**
9. **ip route** *ip-address tunnel num*

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	**interface** *type number* **Example:** `Router(config)# interface tunnel 1`	Specifies the interface type and number and enters interface configuration mode. • To configure a tunnel, use **tunnel** for the type argument.

Step 4	**tunnel mode** {**greip**|**gremultipoint**}
	Example:
	`Router(config-if)# tunnel mode gre ip`
		Specifies the encapsulation protocol to be used in the tunnel.

Step 5	**mpls ip** **Example:** `Router(config-if)# mpls ip`	Enables MPLS on the tunnel.

Step 6	**tunnel source** {*ip-address* | *interface-typeinterface-number*}
	Example:
	`Router(config-if)# tunnel source 1.1.1.1`
		Configures the tunnel source.
		• Use the *ip-address* argument to specify the source IP address.
		• Use the *interface-type* and *interface-number* arguments to specify the interface to use.
		Note The tunnel source and destination IP addresses must be defined on both PE routers.

	Command or Action	Purpose

Step 7	**tunnel destination** {*hostname* | *ip-address*}
	Example:
	`Router(config-if)# tunnel destination 3.3.3.3`
		Configures the tunnel destination.
		• Use the hostname argument to specify the name of the host destination.
		• Use the ip-address argument to specify the IP address of the host destination.
		Note The tunnel source and destination IP addresses must be defined on both PE routers.

Step 8	**end**
	Example:
	`Router(config-if)# exit`
		Exits interface configuration mode and returns to privileged EXEC mode.

Step 9	**ip route** *ip-address tunnel num* **Example:** `Router(config)# ip route 10.2.2.2 255.255.255.255 Tunnel1` **Example:**	Creates a static route.

Configuration Examples for L2VPN Advanced VPLS

The following sections show configuration examples for the three supported methods of configuring the L2VPN Advanced VPLS feature

- Configuring L2VPN Advanced VPLS—Explicitly Specifying Peer PE Routers Example, page 12
- Configuring L2VPN Advanced VPLS—Using MPLS Traffic Engineering Tunnels Example, page 13
- Configuring L2VPN Advanced VPLS—Using MPLS over GRE Tunnels Example, page 13

Configuring L2VPN Advanced VPLS—Explicitly Specifying Peer PE Routers Example

The following example shows how to create two VPLS domains under VLANs 10 and 20. Each VPLS domain includes two pseudowires to peer PE routers 10.2.2.2 and 10.3.3.3. Load-balancing is enabled through the l**oad-balanceflow** and **flow-labelenable** commands.

```
pseudowire-class cl1
   encap mpls
   load-balance flow
   flow-label enable
!
port-channel load-balance src-mac
!
interface virtual-ethernet 1
   transport vpls mesh
      neighbor 10.2.2.2 pw-class cl1
      neighbor 10.3.3.3 pw-class cl1
   switchport
   switchport mode trunk
   switchport trunk allowed vlan 10, 20
```

Configuring L2VPN Advanced VPLS—Using MPLS Traffic Engineering Tunnels Example

The following example shows the creation of two VPLS domains and uses MPLS Traffic Engineering tunnels to specify the explicit path.

```
pseudowire-class cl1
   encap mpls
!
port-channel load-balance src-mac
!
interface Tunnel1
   ip unnumbered Loopback0
   tunnel mode mpls traffic-eng
   tunnel destination 192.168.1.1
   tunnel mpls traffic-eng autoroute announce
   tunnel mpls traffic-eng path-option 1 explicit name LSP1
!
ip explicit-path name LSP1 enable
   next-address 192.168.2.2
   next-address loose 192.168.1.1
!
interface Tunnel2
   ip unnumbered Loopback0
   tunnel mode mpls traffic-eng
   tunnel destination 172.16.1.1
   tunnel mpls traffic-eng autoroute announce
   tunnel mpls traffic-eng path-option 1 explicit name LSP2
!
ip explicit-path name LSP2 enable
   next-address 172.16.2.2
   next-address loose 172.16.1.1
!
interface virtual-ethernet 1
    transport vpls mesh
      neighbor 10.2.2.2 pw-class cl1
      neighbor 10.3.3.3 pw-class cl1
   switchport
   switchport mode trunk
   switchport trunk allowed vlan 10,20
```

Configuring L2VPN Advanced VPLS—Using MPLS over GRE Tunnels Example

The following example shows the creation of two VPLS domains under VLANs 10 and 20. Each VPLS domain includes two pseudowires to peer PEs 10.2.2.2 and 10.3.3.3. The pseudowires are MPLS over GRE tunnels because the core is IP.

```
pseudowire-class cl1
   encap mpls
   load-balance flow
!
```

```
port-channel load-balance src-mac
!
int tunnel 1
   tunnel mode gre ip
   mpls ip
   tunnel source 10.1.1.1
   tunnel destination 10.2.2.2
!
int tunnel 2
   tunnel mode gre ip
   mpls ip
   tunnel source 10.1.1.1
   tunnel destination 10.3.3.3
!
interface virtual-ethernet 1
   transport vpls mesh
      neighbor 10.2.2.2 pw-class cl1
      neighbor 10.3.3.3 pw-class cl1
   switchport
   switchport mode trunk
   switchport trunk allowed vlan 10, 20
ip route 10.2.2.2 255.255.255.255 Tunnel1
ip route 10.2.2.2 255.255.255.255 Tunnel2
```

Additional References

The following sections provide references related to the L2VPN Advanced VPLS feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
VPLS	Cisco 7600 Series Ethernet Services Plus (ES+) and Ethernet Services Plus T (ES+T) Line Card Configuration Guide
MPLS Traffic Engineering tunnels	MPLS Traffic Engineering and Enhancements
GRE tunnels	Implementing Tunnels
Cisco 6500 LAN ports	Configuring LAN Ports for Layer 2 Switching

Standards

Standard	Title
draft-bryant-filsfils-fat-pw	I-D: Flow Aware Transport of MPLS Pseudowires (FAT PWs)

MIBs

MIB	MIBs Link
• N/A	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
RFC 4762	Virtual Private LAN Services (VPLS) Using Label Distribution Protocol (LDP) Singling

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/cisco/web/support/index.html

Feature Information for L2VPN Advanced VPLS

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 L2VPN Advanced VPLS

Feature Name	Releases	Feature Information
L2VPN Advanced VPLS	12.2(33)SXI4	L2VPN Advanced VPLS feature uses Virtual Switch System (VSS) and Flow Aware Transport (FAT) pseudowires to achieve PE redundancy and load-balancing. In 12.2(33)SXI4, this feature was introduced on the Cisco 6500 series router. The following commands were introduced: **flow-label enable** , **interfacevirtual-ethernet**, **load-balanceflow**, **neighbor(VPLStransportmode)**, **showinterfacevirtual-ethernet**, and **transportvplsmesh**. The following command was modified: **showmplsl2transportvc**